

Lecture Notes in Artificial Intelligence

2499

Subseries of Lecture Notes in Computer Science

Edited by J. G. Carbonell and J. Siekmann

Lecture Notes in Computer Science

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Tokyo

Stephen D. Richardson (Ed.)

Machine Translation: From Research to Real Users

5th Conference of the Association for Machine Translation
in the Americas, AMTA 2002
Tiburon, CA, USA, October 8 – 12, 2002
Proceedings



Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editor

Stephen D. Richardson
Microsoft Research
1 Microsoft Way, Redmond, WA 98052, USA
E-mail: steveri@microsoft.com

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Machine translation: from research to real users : Tiburon, CA, USA, October 8 - 12, 2002 ; proceedings / Stephen D. Richardson (ed.). - Berlin ; Heidelberg ; New York ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002

(... Conference of Association for Machine Translation in the Americas, AMTA ... ; 5)

(Lecture notes in computer science ; Vol. 2499 : Lecture notes in artificial intelligence)

ISBN 3-540-44282-0

CR Subject Classification (1998): I.2.7, I.2, F.4.2-3, I.7.1-3

ISSN 0302-9743

ISBN 3-540-44282-0 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag Berlin Heidelberg New York,
a member of BertelsmannSpringer Science+Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2002
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Markus Richter, Heidelberg
Printed on acid-free paper SPIN: 10870740 06/3142 5 4 3 2 1 0

Preface

AMTA 2002: From Research to Real Users

Ever since the showdown between Empiricists and Rationalists a decade ago at TMI 92, MT researchers have hotly pursued promising paradigms for MT, including data-driven approaches (e.g., statistical, example-based) and hybrids that integrate these with more traditional rule-based components.

During the same period, commercial MT systems with standard transfer architectures have evolved along a parallel and almost unrelated track, increasing their coverage (primarily through manual update of their lexicons, we assume) and achieving much broader acceptance and usage, principally through the medium of the Internet. Webpage translators have become commonplace; a number of online translation services have appeared, including in their offerings both raw and postedited MT; and large corporations have been turning increasingly to MT to address the exigencies of global communication. Still, the output of the transfer-based systems employed in this expansion represents but a small drop in the ever-growing translation marketplace bucket.

Now, 10 years later, we wonder if this mounting variety of MT users is any better off, and if the promise of the research technologies is being realized to any measurable degree. In this regard, the papers in this volume target responses to the following questions:

- Why aren't any current commercially available MT systems primarily data-driven?
- Do any commercially available systems integrate (or plan to integrate) data-driven components?
- Do data-driven systems have significant performance or quality issues?
- Can such systems really provide better quality to users, or is their main advantage one of fast, facilitated customization?
- If any new MT technology could provide such benefits (somewhat higher quality, or facilitated customization), would that be the key to more widespread use of MT, or are there yet other more relevant unresolved issues, such as system integration?
- If better quality, customization, or system integration aren't the answer, then what is it that users really need from MT in order for it to be more useful to them?

The contributors to this volume have sought to shed light on these and related issues from a variety of viewpoints, including those of MT researchers, developers, end-users, professional translators, managers, and marketing experts. The jury appears still to be out, however, on whether data-driven MT, which seems to have meandered along a decade-long path of evolution (instead of revolution, as many thought it might be), will lead us to the holy grail of high-quality MT. And yet, there is a sense of progress and optimism among the practitioners of our field.

I extend my sincere thanks to the members of the AMTA 2002 program committee, who sacrificed time and effort to provide detailed analyses of the papers submit-

ted to the conference. Many of the authors expressed gratitude for the insightful and helpful comments they received from the reviewers as they prepared their papers for publication.

Many thanks also go to the organizers of AMTA 2002 who spent untold hours to ensure the success of the conference:

Elliott Macklovitch, General Chair
Violetta Cavalli-Sforza, Local Arrangements Chair
Robert Frederking, Workshops and Tutorials
Laurie Gerber, Exhibits Coordinator
Jin Yang, Webmaster
Debbie Becker, Registrar

In particular, I am grateful to Elliott and Laurie, who provided me with a constant and sustaining flow of guidance and wisdom throughout the conception and assemblage of the program.

Final and special thanks go to Deborah Coughlin, who assisted me in managing the submissions to the conference and in overseeing all substantial aspects of the production of this volume, and to my other colleagues at Microsoft Research, who supported us during this process.

August 2002

Stephen D. Richardson

Program Committee

Arendse Bernth, IBM T.J. Watson Research Center
Christian Boitet, Pr. Université Joseph Fourier, GETA, CLIPS, IMAG
Ralf Brown, Carnegie Mellon University Language Technologies Institute
Robert Cain, MT Consultant
Michael Carl, Université de Montréal, RALI
Bill Dolan, Microsoft Research
Laurie Gerber, Language Technology Broker
Stephen Helmreich, New Mexico State University Computing Research Laboratory
Eduard Hovy, University of Southern California Information Science Institute
Pierre Isabelle, Xerox Research Centre Europe
Christine Kamprath, Caterpillar Corp.
Elliott Macklovitch, Université de Montréal, RALI
Bente Maegaard, Center for Sprogteknologi
Michael McCord, IBM T.J. Watson Research Center
Robert C. Moore, Microsoft Research
Hermann Ney, RWTH Aachen
Sergei Nirenburg, New Mexico State University Computing Research Laboratory
Franz Och, RWTH Aachen
Joseph Pentheroudakis, Microsoft Research
Jessie Pinkham, Microsoft Research
Fred Popowich, Gavagai Technology Inc.
Florence Reeder, MITRE Corp.
Harold Somers, UMIST
Keh-Yih Su, Behavior Design Corp.
Eiichiro Sumita, ATR
Hans Uszkoreit, Saarland University at Saarbrücken, DFKI
Lucy Vanderwende, Microsoft Research
Hideo Watanabe, IBM Tokyo Research Laboratory
Andy Way, Dublin City University
Eric Wehrli, University of Geneva
John White, Northrop Grumman Information Technology
Jin Yang, SYSTRAN
Ming Zhou, Microsoft Research

Tutorial Descriptions

Example-Based Machine Translation

Ralf Brown

Language Technologies Institute
Carnegie Mellon University
ralf@cs.cmu.edu

1 Description

This tutorial will introduce participants to the history and practice of example-based machine translation (EBMT). After a definition of EBMT and an overview of its origins (Sato and Nagao, among others), various types of approaches to example-based translation (such as deep versus shallow processing) will be presented. This discussion will lead into a overview of a number of recent example-based systems, both "pure" and hybrid systems combining rule-, statistics-, or knowledge-based approaches with EBMT. Candidates for discussion include EGDAR, Gaijin, ReVerb, and systems by Cranias, Guevenier/Cicekli, and Streiter. Finally, the tutorial will conclude with a more in-depth examination of the Generalized EBMT system developed at Carnegie Mellon University.

2 Outline

- Introduction: Example-Based Translation's definition and origins
- EBMT and its relation with other translation technologies
 - the Vaquois diagram and "depth" of processing
 - "shallow" EBMT and translation memories
 - "deep" EBMT and transfer-rule systems
 - relationship between EBMT and statistical MT
- Overview of EBMT Systems
 - EDGAR
 - Gaijin
 - ReVerb
 - etc.
- Hands-On Exercise in EBMT
- Carnegie Mellon University's Generalized EBMT system

- simple matching against an example base
- generalizing the examples into templates
- learning how to generalize
- inexact matching
- use as an engine in the Multi-Engine MT architecture

3 Biographical Information

Ralf Brown has been working on Example-Based Translation since 1995, using it in various applications such as the translation component of a speech-to-speech translation system and for document translation for event tracking in news streams. He received his Ph.D. in Computer Science from Carnegie Mellon University in 1993, where he is currently research faculty in the Language Technologies Institute.

Units of Meaning in Translation — Making Real Use of Corpus Evidence

Pernilla Danielsson (in co-operation with Prof Wolfgang Teubert)

Centre for Corpus Linguistics
 Department of English
 University of Birmingham
 Birmingham B15 2TT
 United Kingdom
 Web-address: www.english.bham.ac.uk/ccl
pernilla@ccl.bham.ac.uk

1 Description

Birmingham's Centre for Corpus Linguistics offers to give a tutorial on how to use large corpora in language research, especially translation. This tutorial will focus on meaning. In modern corpus linguistics, we work from the hypothesis that meaning is in its use, as previously stated by researchers such as Terry Winograd or Wittgenstein in his 'Philosophical Investigations' stating that 'the meaning of a word is in its use'. This may be further interpreted into the statement that meaning is in text, which opposes the idea that meaning is constructed in the human brain. From a research point of view, this is a very important statement since it will remove the difficulties of trying to model human brains in order to interpret a text and instead gear us towards finding new methods of interpreting the complex systems governing the interpretation of texts. This tutorial will show how by carefully examining large corpora, meaning can emerge through patterning.

Units of meaning are often larger and more complex than the simple word. Most units of translation are compounds, collocations or even phrases. As for single words, most of them are

ambiguous. The participants will be shown methods to disambiguate words by investigating their contextual profiles. The tutorial will also focus on retrieving translation equivalents and learning how corpus data can help us produce translated texts that display the ‘naturalness’ of the target language.

2 Outline

The tutorial will cover the following topics:

- Working with Large Corpora
- Units of Meaning
- Translation Units in Parallel texts
- Contextual Profiles

The tutorial will be divided into two sessions in order to cover both monolingual and multilingual corpus methodologies. The first session is focused on the use of methods in monolingual corpus linguistics for extracting information about the languages in question. As a demonstration, we will make use of the Bank of English, a 450 million word corpus co-owned by the University of Birmingham and the publishing house HarperCollins. Once Units of Meaning have been established in the monolingual corpus we will move on to parallel texts. By using the newly discovered units of meaning, we will now discover that words are not as ambiguous as they first seem, at least not when treated within larger units.

3 Biographical Information

Dr. Pernilla Danielsson is the Deputy Director at the Centre for Corpus Linguistics, University of Birmingham. In 2000 she became the new Project Manager for the EU-funded project TELRI-II (Trans European Language Resources Infrastructure). With a background in Computational Linguistics, she is now focusing her research on the study of units of meaning in corpora.

Prof. Wolfgang Teubert (PhD Heidelberg 1979) was, until 2000, a senior research fellow at the Institut für Deutsche Sprache (IDS), Mannheim, Germany. In 2000, he was appointed to the Collins Chair of Corpus Linguistics, Department of English at the University of Birmingham. The focus of his research is the derivation of linguistic metadata from digital resources, particularly in multilingual environments with the emphasis on semantics. His other interest is the application of the methodology of corpus linguistics to critical discourse analysis. He is also the editor of the *International Journal of Corpus Linguistics*.

Supporting a Multilingual Online Audience

Laurie Gerber

On Demand Translation
61 Nicholas Road
Suite B3
Framingham, MA 01701
508-877-3430
lgerber@on-demand.biz

1 Description

The need to provide customer service and technical support to non-English-speaking customers is growing rapidly: Internet users are increasingly multilingual. IDC has reported that the number of Internet users in Western Europe surpassed users in the U.S. at the end of 2001. The trend is expected to continue, and English will eventually become a minority language on the Internet. The growth of the Internet overseas in turn tends to fuel sales of hardware and software applications. But companies that release localized products outside of the U.S. are often unprepared to fully support their increasingly diverse customer base. Simply translating web sites is rarely adequate. Customer communications are conducted through a variety of channels, and may contain very different types of text, speech and data. Translation products and services abound, but understanding which solutions can effectively address a specific need requires an understanding of the problem, as well as the solutions.

This tutorial will provide participants with:

- Practical skills for analyzing language support requirements
- Strategies for selecting and deploying appropriate language solutions
- An understanding of the range and capabilities of language technologies
- Knowledge of how to integrate language technologies into organizational workflows while avoiding common pitfalls
- Suggestions for measuring the effectiveness of your multilingual customer support

2 Outline

- Language products, services, solutions and their applications
- Assessing multilingual communication needs
 - Source materials factors
 - Target materials factors

- Delivery factors
- Organizational factors
- Cost factors
- Implementation factors
- Integrating multiple language technologies to work together
- Preparing your content
- Communicating with customers about language technology
- Measuring results

3 Biographical Information

With a background in Asian languages, Laurie Gerber was a central figure in SYSTRAN Software's Chinese-English and Japanese-English machine translation development efforts from 1986 to 1998 and served as Director of R&D from 1995 through 1998. Also in contact with users, Ms. Gerber developed a strong interest in usability issues for language technology. After earning a Master of Computational Linguistics from the University of Southern California in May 2001, she now works as an independent consultant on language technology implementation, and business development for commercializable prototype language technologies. She is currently Vice President (2000-2002) of the Association for Machine Translation in the Americas, and editor of Machine Translation News International, the newsletter of the International Association for Machine Translation.

The State of the Art in Language Modeling

Joshua Goodman

Machine Learning and Applied Statistics Group
Microsoft Research

<http://www.research.microsoft.com/~joshuago>
JoshuaGo@microsoft.com

1 Description

The most popular approach to statistical machine translation is the source-channel model; language models are the "source" in source-channel. Language models give the probability of word sequences. In a machine translation system they can assist with word choice or word order-- "The flesh is willing" instead of "The meat is willing" or "Flesh the willing is." Most use of language models for statistical MT has been limited to models used by speech recognition systems (trigrams), despite the existence of many other techniques. In addition, many language modeling techniques could be adapted to improve channel models, or other parts of MT systems.

This tutorial will cover the state of the art in language modeling. The introduction will include what a language model is, a quick review of elementary probability, and applications of language modeling, with an emphasis on statistical MT. The bulk of the talk will describe current techniques in language modeling, including techniques like clustering and smoothing that are useful in many areas besides language modeling, and more language-model specific techniques such as high order n-grams and sentence mixture models. Finally, we will describe available toolkits and corpora.

A portion of the material in this talk was developed by Eugene Charniak.

2 Outline

- Introduction quickly reviewing key concepts in probability needed for language models, and then briefly the source-channel model for machine translation. Outline of remainder of talk describing specific language modeling techniques.
- Smoothing addresses the problem of data sparsity: there is rarely enough data to accurately estimate the parameters of a language model. Smoothing gives a way to combine less specific, more accurate information with more specific, but noisier data. I will describe two classic techniques -- deleted interpolation and Katz (or Good-Turing) smoothing -- and one recent technique, Modified Kneser-Ney smoothing, the best known.
- Caching is a widely used technique that uses the observation that recently observed words are likely to occur again. Models from recently observed data can be combined with more general models to improve performance.
- Skipping models use the observation that even words that are not directly adjacent to the target word contain useful information.
- Sentence-mixture models create separate models for different sentence types. Modeling each type separately improves performance.
- Clustering is one of the most useful language modeling techniques. Words can be grouped together into clusters through various automatic techniques; then the probability of a cluster can be predicted instead of the probability of the word.
- There are many recent, but more speculative language modeling techniques, including grammar-based language models, maximum entropy models, and whole sentence maximum entropy models.

- Finally, I will also talk about some practical aspects of language modeling. I will describe how freely available, off-the-shelf tools can be used to easily build language models, where to get data to train a language model, and how to use methods such as count cutoffs or relative-entropy techniques to prune language models.

Those who attend the tutorial should walk away with a broad understanding of current language modeling techniques, and the background needed to either build their own language models, or to apply some of these techniques to machine translation.

3 Biographical Information

Joshua Goodman is a Researcher at Microsoft Corporation. He previously worked on speech recognition at Dragon Systems. He received his Ph.D. in 1998 from Harvard University for work on statistical natural language processing. He then worked in the speech group at Microsoft Research, with a focus on language modeling. Recently, he moved to the Microsoft Research Machine Learning and Applied Statistics group, where he has worked on probabilistic models for natural language tasks, such as grammar checking.

Beyond the Gist: A Post-Editing Primer for Translation Professionals

Walter Hartmann

MTConsulting Co.
wh@mtconsult.com

1 Description

To keep up with ever-increasing demands on output and turn-around, the professional translator needs tools that help increase productivity. Among others, machine translation stands out as a timesaving tool that can greatly enhance translation output. It does take, however, preparation and practice to weave machine translation efficiently into the workflow.

This tutorial addresses the various topics to be considered when contemplating the integration of machine translation into the workflow for publication-ready translations.

2 Outline

The main topics to be covered are:

- Introduction: MT and the professional translator
- Evaluation techniques
 - Application variations
 - Analysis of MT output for a given task
 - Remedies: Pre-editing, dictionary updates, post-editing, when to pass on post-editing
- Editing techniques – hands-on practice using texts translated from various languages into English

3 Biographical Information

MT has been a valuable productivity tool for the presenter since 1985, when he began to post-edit publication-level texts for customers. When MT programs became available for PCs, he pioneered MT integration into the workflow of translation companies. In subsequent years, he integrated machine translation for high-volume translations for several other companies. Even today, as a part-time freelance translator, the presenter actively uses MT as a productivity tool.

MT Evaluation: The Common Thread

John S. White¹ and Florence Reeder²

¹Northrop Grumman Information Technology
McLean, VA USA
white_john@prc.com

²The MITRE Corporation
McLean, VA USA
freeder@mitre.org

1 Description

MT Evaluation is central to the goals of both the research and product worlds, and so can form a common, uniting thread among the two. Useful, comparable measures are difficult to produce in both spheres, despite the lessons from the 1960's. This is due in part to the uniquely difficult aspects of evaluating translation in general. This tutorial sets out to explain some issues, structures, and approaches that will demystify evaluation, and make it possible to design and perform meaningful evaluations with a minimum of time and resource.

The tutorial will cover the topics of the difficulty of MT evaluation, and then presents different views (including the recent work from the ISLE project) on the stakeholders, uses, and types of MT, and the attributes, measurands, and metrics implied by these perspectives. We will present a number of historical methods which may have renewed usefulness in today's context, as well as some approaches over the last decade and new approaches modeled in the last year.

In particular we will address the potential for automatic evaluation of MT. The multiplicity of uses, users, and approaches to MT have traditionally made this a presumptively impossible goal for all of the evaluation metrics. However, new research in evaluation has shed light on the potential for the automatic prediction of certain key attributes of MT output which can be used to predict more general performance. Among the promising approaches for automation include capturing the perspectives of language learning; modeling "machine English" and human English; and predicting fidelity from intelligibility.

The tutorial will provide the participant with perspectives, tools, and data for determining the usefulness of particular approaches in particular MT contexts. The tutorial will be quite interactive, with exercises and MT data to help the participant understand the challenges and potential for MT evaluation.

2 Outline

- Evaluation overview
- What's hard about MT evaluation
- The stakeholders, purposes, and types of MT evaluation
 - Methods related to purposes
 - ISLE classification
- Some famous evaluation methods, old and new
- The search for automatic MT Evaluation
 - New methods
 - Rationales and Bases
 - Interactive Experiments
- Conclusions

3 Biographical Information

John White is Director of Independent Research and Development for Defense Enterprise Solution, Northrop Grumman Information Technology. In this capacity he is responsible for research and development initiatives in language systems evaluation, information assurance, software agent technology, modeling/simulation, collaborative computing, and imaging.

White holds a Ph.D. in Linguistic Anthropology from The University of Texas, and is widely published in machine translation, evaluation, artificial intelligence, and information assurance.

Florence Reeder is an Associate Technical Area Manager with The Mitre Corporation. She works with a variety of U.S. government agencies in developing machine translation and foreign language handling systems which multiply scarce expertise in critical languages. Florence is also a doctoral student at George Mason University, and is currently completing her dissertation work in the field of MT evaluation.

Table of Contents

Technical Papers

Automatic Rule Learning for Resource-Limited MT	1
<i>Jaime Carbonell, Katharina Probst, Erik Peterson, Christina Monson, Alon Lavie, Ralf Brown, and Lori Levin</i>	
Toward a Hybrid Integrated Translation Environment	11
<i>Michael Carl, Andy Way, and Reinhard Schler</i>	
Adaptive Bilingual Sentence Alignment	21
<i>Thomas C. Chuang, GN You, and Jason S. Chang</i>	
DUSTer: A Method for Unraveling Cross-Language Divergences	31
for Statistical Word-Level Alignment <i>Bonnie J. Dorr, Lisa Pearl, Rebecca Hwa, and Nizar Habash</i>	
Text Prediction with Fuzzy Alignments	44
<i>George Foster, Philippe Langlais, and Guy Lapalme</i>	
Efficient Integration of Maximum Entropy Lexicon Models within	54
the Training of Statistical Alignment Models <i>Ismael Garc�a-Varea, Franz J. Och, Hermann Ney, and Francisco Casacuberta</i>	
Using Word Formation Rules to Extend MT Lexicons	64
<i>Claudia Gdaniec and Esm� Manandise</i>	
Example-Based Machine Translation via the Web	74
<i>Nano Gough, Andy Way, and Mary Hearne</i>	
Handling Translation Divergences: Combining Statistical and Symbolic	84
Techniques in Generation-Heavy Machine Translation <i>Nizar Habash and Bonnie Dorr</i>	
Korean-Chinese Machine Translation Based on Verb Patterns	94
<i>Changhyun Kim, Munpyo Hong, Yinxia Huang, Young Kil Kim, Sung Il Yang, Young Ae Seo, and Sung-Kwon Choi</i>	
Merging Example-Based and Statistical Machine Translation: An Experiment ...	104
<i>Philippe Langlais and Michel Simard</i>	
Classification Approach to Word Selection in Machine Translation	114
<i>Hyo-Kyung Lee</i>	

Better Contextual Translation Using Machine Learning 124
Arul Menezes

Fast and Accurate Sentence Alignment of Bilingual Corpora 135
Robert C. Moore

Deriving Semantic Knowledge from Descriptive Texts Using an MT System 145
*Eric Nyberg, Teruko Mitamura, Kathryn Baker, David Svoboda,
Brian Peterson, and Jennifer Williams*

Using a Large Monolingual Corpus to Improve Translation Accuracy 155
Radu Soricut, Kevin Knight, and Daniel Marcu

Semi-automatic Compilation of Bilingual Lexicon Entries from 165
Cross-Lingually Relevant News Articles on WWW News Sites
Takehito Utsuro, Takashi Horiuchi, Yasunobu Chiba, and Takeshi Hamamoto

Bootstrapping the Lexicon Building Process for Machine Translation 177
between ‘New’ Languages
Ruvan Weerasinghe

User Studies

A Report on the Experiences of Implementing an MT System for Use 187
in a Commercial Environment
Anthony Clarke, Elisabeth Maier, and Hans-Udo Stadler

Getting the Message In: A Global Company’s Experience with the New 195
Generation of Low-Cost, High Performance Machine Translation Systems
Verne Morland

An Assessment of Machine Translation for Vehicle Assembly Process 207
Planning at Ford Motor Company
Nestor Rychtyckyj

System Descriptions

Fluent Machines’ EliMT System 216
Eli Abir, Steve Klein, David Miller, and Michael Steinbaum

LogoMedia TRANSLATE™, version 2.0 220
Glenn A. Akers

Natural Intelligence in a Machine Translation System	224
<i>Howard J. Bender</i>	
Translation by the Numbers: Language Weaver	229
<i>Bryce Benjamin, Kevin Knight, and Daniel Marcu</i>	
A New Family of the PARS Translation Systems	232
<i>Michael Blekhman, Andrei Kursin, and Alla Rakova</i>	
MSR-MT: The Microsoft Research Machine Translation System	237
<i>William B. Dolan, Jessie Pinkham, and Stephen D. Richardson</i>	
The NESPOLE! Speech-to-Speech Translation System	240
<i>Alon Lavie, Lori Levin, Robert Frederking, and Fabio Pianesi</i>	
The KANTOO MT System: Controlled Language Checker and Lexical	244
Maintenance Tool	
<i>Teruko Mitamura, Eric Nyberg, Kathy Baker, Peter Cramer,</i>	
<i>Jeongwoo Ko, David Svoboda, and Michael Duggan</i>	
Approaches to Spoken Translation	248
<i>Christine A. Montgomery, Naicong Li</i>	
Author Index	253

Automatic Rule Learning for Resource-Limited MT

Jaime Carbonell, Katharina Probst, Erik Peterson, Christian Monson,
Alon Lavie, Ralf Brown, and Lori Levin

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213

{jgc, kathrin, eepeter, cmonson, alavie, ralf, lsl}@cs.cmu.edu

Abstract. Machine Translation of minority languages presents unique challenges, including the paucity of bilingual training data and the unavailability of linguistically-trained speakers. This paper focuses on a machine learning approach to transfer-based MT, where data in the form of translations and lexical alignments are elicited from bilingual speakers, and a seeded version-space learning algorithm formulates and refines transfer rules. A rule-generalization lattice is defined based on LFG-style f-structures, permitting generalization operators in the search for the most general rules consistent with the elicited data. The paper presents these methods and illustrates examples.

1 Introduction

Machine Translation (MT) for minority languages offers significant potential benefits, but also presents serious scientific and economic challenges. Among the benefits are: communication between isolated, often economically disadvantaged groups (i.e. indigenous groups in Latin America) and the speakers of majority languages, and the potential preservation of endangered languages – over half of the 6,000 presently existing languages worldwide. The primary scientific challenge is the creation of MT systems for languages of little economic importance at very low cost per language, including the acquisition of linguistic information with minimal pre-existing bilingual corpora and little or no previous linguistic analysis of the minority language.

In order to address these needs, we are investigating omnivorous MT systems, including statistical and example-based MT when some parallel training corpora can be acquired, and machine learning of transfer-based MT rules when access to a native non-linguist informant permits partial elicitation of linguistic information, such as translations of model sentences and lexical-level bilingual alignments. This paper focuses on this last objective of our project: supervised learning of transfer rules with the aid of an elicitation interface to a bilingual native speaker without any assumptions regarding his or her linguistic sophistication. While our technology is eventually aimed at low-density languages, it is intended to be target language independent. Hence, we are developing the

system using examples from various languages, such as Chinese, German, Mapudungun¹, and Swahili. For illustration purposes, we present examples in these languages throughout the paper.

2 Elicitation Corpus

Since there are usually little or no bilingual translated data available for minority languages, we elicit the minimal amount of information required from a bilingual informant. The informant translates a set of constituents or sentences constructed prior to learning – the same set for all languages – designed to elicit all the basic grammatical constructions, based on the principles of field linguistics. The informant supplies lexical alignments as well as translations from the source language (SL) into the target language (TL). Currently, the source language is English or Spanish.

The elicitation corpus contains sentences and sub-sentential constituents carefully constructed to obtain information about typologically prevalent features (such as tense and number) and their possible values in the target language (such as singular, dual, plural). To this end, we rely on research done in field linguistics, and use lists that were designed for this task, such as [2] and [1]. The corpus exploits compositionality, starting with smaller constituents and recycling them in higher level ones, paralleling the compositional machine learning for version spaces discussed below.

In designing the corpus, we strive to maximize coverage while minimizing size by targeting specific constructions. For instance, in order to infer rules for relative clauses, we developed a set of sentences that exhibit different types of relative clauses, such as subject and object relative clauses. If an uncontrolled corpus was used, it would need to be several orders of magnitude larger to cover a comparable range of linguistic variability, and therefore it would impose a much larger burden on the bilingual informant. Further, a controlled corpus allows us to plan a process where the informant only translates part of the corpus, and leaves other parts that have been determined to be irrelevant for the given target language untranslated. For more detail on the elicitation process, please refer to [7] and [8].

3 Translation Engine

Rule-based translation is done through a custom-built translation engine using an integrated transfer approach with analysis, transfer, and generation stages. In the engine, as in the early METAL system [4], each analysis rule is also coupled with a corresponding transfer action, as enforced by the transfer rule formalism. Table 1 provides examples: Comments are indicated by one or more semicolons. The first line contains the SL and TL category, followed by the constituent sequences.

¹ Mapudungun, spoken in Chile, is one of the minority languages we focus on.

The formalism is able to handle a variety of common translation divergences, including head-switching, changes in grammatical relations such as an object in the source language being expressed as a subject in the target language, structural changes such as having an NP become a PP in another language, and lexical gaps where one target word replaces an entire source phrase [9].

Table 1. Sample Transfer Rules

Rule to handle non-auxiliary verb question transfer from Chinese to English	Sample lexical rule 1	Sample lexical rule 2
S::S : [NP VP MA] → [V NP VP "?"] (;;alignments: (x1::y2) (x2::y3) ;;x-side constraints: ;; (parsing) ((x0 subj) == x1) ((x0 subj case) = nom) (x0 = x2) ((x0 act) = *quest) ;;xy-constraints: ;; (transfer) (y0 = x0) ((y0 act) =c *quest) ;;y-side constraints: ;; (generation) ((y1 form) = *do) ((y1 agr) = (y2 agr)) (y2 == (y0 subj)) (y3 = y0))	AUX::AUX : [zuo4] → [do] (;;alignments: ((x1::y1) ;;x-side constraints: ;; (parsing) ;;xy-side constraints: ;; (transfer) ;;y-side constraints: ((Y0 form) = *do) ((y0 agr) = (*or* *1-sg *2-sg *plu)) ((y0 tense) = *pres))	AUX::AUX : [zuo4] → [does] (;;alignments: (x1::y1) ;;x-side constraints: ;; (parsing) ;;xy-constraints: ;; (transfer) ;;y-side constraints: ((y0 form) = *do) ((y0 agr) = *3-sg) ((y0 tense) = *pres))

Translation starts with a bottom-up unification-based chart parser that produces an ambiguity-packed representation of the source language input. If at any one point more than one rule can be applied to a structure, the rules are applied in the order in which they appear in the grammar. Analysis builds both a syntactic constituent (or for short c-)structure and an associated feature (f-)structure. For example, in the structural transfer rule in figure 1, which handles the transfer of some types of interrogative sentences from Mandarin to English, the feature structure for the source language (SL) S constituent, represented by x0, is built using the x-side, i.e. parsing, constraints.

Transfer is done from the top down, starting at the top node in the chart created during parsing. Transfer rules explicitly state the alignment of source and target constituents through equations such as (x1::y2) and (x2::y3) in the

example rule, indicating that the first source constituent maps to the second target constituent, and the second source constituent maps to the third target constituent. Not all source constituents need to map to the target (such as the Chinese question particle MA at x3 in this rule which is deleted during transfer). Target constituents that are not aligned to a source constituent will be created based upon the feature structure assigned to them in the transfer and generation equations. During transfer, the engine uses these alignments to reorder constituents.

Features may also be passed from source to target using xy-constraints. For example, the $(y0 = x0)$ equation in table 1 copies the entire source sentence feature structure to the target.

During generation, the target side feature structures are built using the y-side constraints. In the example rule, the first constituent V (y1) has its form set to ‘do’, which is used later to choose the correct verb to insert. To enforce that the form of ‘do’ agrees with the subject NP at y2, the $((y1 \text{ agr}) = (y2 \text{ agr}))$ is used.

Once a selected transfer rule is applied, transfer and generation continues by recursively running applicable transfer rules on the sub-constituents of the rule until the lexical level is reached. Lexical transfer rules then apply to select the appropriate translation. In the example, the auxiliary ‘do’ is inserted at the start of the sentence. The appropriate form of ‘do’ is selected based upon the agreement constraint between the subject NP and initial V in the sentence rule and the agreement features in the individual lexical rules.

4 Seeded Version Spaces for Transfer Rule Learning

Transfer rule learning consists of three steps. The first step, feature detection, determines what linguistic features are present in the target language. For instance, does the language have a number-agreement feature? If so, is the distinction between singular and plural, or between singular, dual and plural? The details of feature detection are beyond the scope of this paper; the interested reader should refer to [7]. The second step is seed-rule generation, where preliminary transfer rules are generated based on lexically-aligned translated constituents. Seed generation produces transfer rules that are very specific to the training examples, essentially defining the operational *S-boundary* (specific boundary) for Version-Space learning [6] [5] [3]. The third step, Seeded Version Space Learning, generalizes the seed rules in order to produce transfer rules that can be used to translate larger classes of unseen examples.

4.1 Seed Generation

Version Space learning suffers worst-case exponential behavior as a function of the number of generalization (or specialization) steps required from initial instances. But if the initial instances are already generalized and the number of

additional generalization steps k -bounded, or limited to a greedy-search, the process is at worst a k -degree polynomial. Hence, to ensure computational tractability, we first build generalized rule-instances called ‘seed rules,’ to the extent that we can build first-level, error-free deterministic generalization. Future work will relax the error-free assumption by substituting a retractable S -boundary.

Seed rules represent generalizations over the lexical level, but are still very specific to the lexically-aligned sentence pair from which they are derived. The major-language (English) sentences in the training corpus have been pre-parsed and disambiguated in advance, so that the system has access to a correct f -structure and a correct phrase or c -structure for each sentence. The seed generation algorithm can proceed in different settings based on how much information is available about the target language. In one extreme case, we assume that we have a fully-inflected target language dictionary. The other extreme is the absence of any information, in which case the system makes assumptions about what linguistic feature values transfer to the target language. For example, we assume that number and definiteness transfer across languages, but not gender or case. This information is either defaulted or extracted in the earlier feature detection phase.

During seed generation, all transfer rule components are produced from the sentence pair, with the exception of xy -constraints.² Table 2 below summarizes how each part of the seed rule is produced.

Table 2. Summary of Seed Generation

SL part-of-speech sequence	English c -structure
TL part-of-speech sequence	1) from the TL dictionary 2) from the English POS sequence
Alignments	from the user
x-side constraints	from the English f -structure
y-side constraints	1) from the TL dictionary 2) from the corresponding English words
xy-constraints	Not produced during seed generation

To make these concepts more concrete, consider an example of a seed rule as produced by our algorithm for simple NPs in English-German (for expository simplicity), as illustrated in table 4 in section 4.3: The English part-of-speech sequence was obtained from the English parse. The German part-of-speech sequence was obtained from a combination of the target language dictionary and the assumption that in the absence of unambiguous target language information, parts of speech transfer into the target language for aligned words. The lexical alignments were given by the bilingual informant. The x -side constraints were read off the English c - and f -structures. The y -side constraints were again obtained by a combination of target language information and ‘safe’ transfer

² xy -constraints are inferred during the Seeded Version Space Learning.

heuristics. If feature information is given in the target language, it is used. If no such information is available, features such as agreement and count project their values onto the target language. This means that target language words obtain their feature values from their corresponding words in the source language.

4.2 Compositionality

In order to scale to more complex examples, the system learns compositional rules: when producing a rule e.g. for a sentence, we can make use of sub-constituent transfer rules already learned. Then, the new seed rule (the higher-level rule) is no longer a flat rule, but consists of previously learned transfer constituents (lower-level rules) such as NPs or ADJPs.

Compositional rules are learned by first producing a flat rule as was described above from the bilingual constituent (phrase or sentence). In the following, the system traverses the c-structure parse of the English constituent. Each node in the parse is annotated with a label such as NP, and roots a subtree, which itself covers part of the constituent. The system then checks if there exists a lower-level rule that can be used to correctly translate the words in this subtree. Consider the following example:

```
(<NP> (<DET> ((ROOT *THE)) THE)
(<NBAR>
(<ADJP> (<ADVP> (<ADV> ((ROOT *VERY)) VERY))
(<ADJP> (<ADJ> ((ROOT *TALL)) TALL)))
(<NBAR> (<N> ((ROOT *WOMAN)) WOMAN)))
```

This c-structure represents the parse tree for the NP ‘the very tall woman’. The system traverses the c-structure from the top in a depth-first fashion. For each node, such as NP, NBAR, etc., it extracts the part of the constituent that is rooted at this node. For example, the node ADJP covers the chunk ‘very tall’. The question is now if there already exists a learned transfer rule that can correctly translate this chunk. To determine what the reference translation should be, the system consults the user-specified alignments and the given target language translation of the training example. In this way, it gains access to a bilingual sentence chunk. In our example the bilingual sentence chunk is ‘very tall’ - ‘sehr grosse’. It also obtains its SL category, in this example ADJP.

In the following, it calls the transfer engine with this chunk, and is returned zero or more translations, together with the f-structures that are associated with each translation. In case there exists a transfer rule that can translate the sentence chunk in question, the system takes note of this and traverses the rest of the c-structure, excluding the chunk’s subtree. After this step is completed, the original flat transfer rule is modified to reflect any compositionality that was found. The rightmost column in table 3 presents an example of how the flat rule is modified to reflect the existence of an ADJP rule that can translate the chunk

Table 3. Example of Compositionality

Lower-level rule	Uncompositional Rule	Compositional Rule
::SL: very tall	::SL: the very tall woman	::SL: the very tall woman
::TL: sehr gross	::TL: die sehr grosse Frau	::TL: die sehr grosse Frau
ADJP::ADJP	NP::NP	NP::NP
[ADV ADJ]	[DET ADV ADJ N]	[DET ADJP N]
→ [ADV ADJ]	→ [DET ADV ADJ N]	→ [DET ADJP N]
(;alignments:	(;alignments:	(;alignments:
(x1::y1)	(x1::y1)	(x1::y1)
(x2::y2)	(x2::y2)	(x2::y2)
	(x3::y3)	(x3::y3)
	(x4::y4)	
::x-side constraints:	::x-side constraints:	::x-side constraints:
	((x1 agr) = *3-sing)	((x1 agr) = *3-sing)
	((x1 def) = *def)	((x1 def) = *def)
	((x2 agr) = *3-sing)	
	((x2 count) = +)	
	((x3 agr) = *3-sing)	((x3 agr) = *3-sing)
	((x4 agr) = *3-sing)	((x3 count) = +)
	((x4 count) = +)	((x3 count) = +)
(x0 = x2)	(x0 = x4)	(x0 = x3)
::y-side constraints:	::y-side constraints:	::y-side constraints:
	((y1 gender) = *f)	((y1 gender) = *f)
	((y1 case) = (*or* *acc *nom))	((y1 case) = (*or* *acc *nom))
	((y1 agr) = *3-sing)	((y1 agr) = *3-sing)
	((y1 def) = *def)	((y1 def) = *def)
	((y3 gender) = *f)	((y2 gender) = *f)
	((y3 case) = (*or* *acc *nom))	((y2 case) = *nom)
	((y3 def) = *def)	((y2 def) = *def)
	((y3 agr) = *3-sing)	((y2 agr) = *3-sing)
	((y4 gender) = *f)	((y3 gender) = *f)
	((y4 case) = (*or* *acc *nom))	((y3 case) = (*or* *acc *nom))
	((y4 agr) = *3-sing)	((y3 agr) = *3-sing)
	((y4 count) = +)	((y3 count) = +)
(y0 = y2)	(y0 = y4)	(y0 = y3)

‘very tall’. The flat, uncompositional rule can be found in the middle column, whereas the lower-level ADJP rule can be found in the leftmost column.

First, the part-of-speech sequence from the flat rule is turned into a constituent sequence on both the SL and the TL sides, where those chunks that are translatable by lower-level rules are represented by the category information of the lower-level rule, in this case ADJP. The alignments are adjusted to the new sequences. Lastly, the constraints must be changed. The x-side constraints are mostly retained (with the indices adjusted to the new sequences). However, those constraints that pertain to the sentence/phrase part that is accounted for by the lower-level rule are eliminated. In the example in table 3, all the x-side constraints on the indices x2 and x3 are removed.

Finally, the y-side constraints are adjusted. For each sentence chunk that was correctly translated by a lower-level rule, the compositionality module compares the f-structures of the correct translation and the incorrect translations as returned by the transfer engine. This is done so as to determine what constraints need to be added to the higher level rule in order to produce the correct translation in context. For each constraint in the correct translation, the system checks if this constraint appears in *all* other translations. If this is not the case, a new constraint is constructed and inserted into the compositional rule. Before simply inserting the constraint, however, the indices need to be adjusted to the higher-level constituent sequence, as can again be seen in the example in table 3.

4.3 Seeded Version Space Learning

The first step in Seeded Version Space Learning is to group the seed rules by their constituent sequences, alignments, and category information. This means that in each group the seed rules differ only in the constraints. The learning algorithm is run on each group separately, as each group corresponds to a target concept (i.e. a target generalized transfer rule), thus it defines a version space. At the heart of the version space learning is the merging of two transfer rules to a more general transfer rule. To this end, it is necessary to clearly define the partial order by which the generality of transfer rules can be assessed (i.e. how the implicit generalization lattice is constructed):

Definition 1 *A transfer rule tr_1 is strictly more general than another transfer rule tr_2 if all f-structures that are satisfied by tr_2 are also satisfied by tr_1 . The two transfer rules are equivalent if and only if all f-structures that are satisfied by tr_1 are also satisfied by tr_2 .*

Based on this definition, we can define operations that will turn a transfer rule tr_1 into a strictly more general transfer rule tr_2 . In particular, we identified three generalization operations:

1. Deletion of a value constraint
2. Deletion of an agreement constraint
3. Merging of two value constraints into one agreement constraint. Two value constraints can be merged if they are of the following format:

$$\begin{aligned} &((X_i \text{ feature}_k) = value_i) \\ &((X_j \text{ feature}_k) = value_i) \\ &\rightarrow ((X_i \text{ feature}_k) = (X_j \text{ feature}_k)), \end{aligned}$$

or similarly for y-side and xy-constraints.

Generalization is achieved by merging transfer rules, which in turn is based on the three generalization operations defined above. Suppose we wish to merge two transfer rules tr_1 and tr_2 to produce the most specific generalization of the two, stored in tr_{merged} . The algorithm proceeds in three steps:

Table 4. Seed Rules and Generalized Transfer Rule

SeedRule1	SeedRule2	Generalized Rule
;;SL: the man ;;TL: der Mann NP::NP [DET N] → [DET N] (;alignments: (x1::y1) (x2::y2) ;;x-side constraints: ((x1 agr) = *3-sing) ((x1 def) = *def) ((x2 agr) = *3-sing) ((x2 count) = +) ;;y-side constraints: ((y1 agr) = *3-sing) ((y1 case) = *nom) ((y1 def) = *def) ((y2 gender) = *m) ((y2 agr) = *3-sing) ((y2 case) = *nom) ((y2 gender) = *m)	;;SL: the woman ;;TL: die Frau NP::NP [DET N] → [DET N] (;alignments: (x1::y1) (x2::y2) ;;x-side constraints: ((x1 agr) = *3-sing) ((x1 def) = *def) ((x2 agr) = *3-sing) ((x2 count) = +) ;;y-side constraints: ((y1 agr) = *3-sing) (y1 case) = (*not* *gen *dat)) ((y1 def) = *def) ((y2 gender) = *f) ((y2 agr) = *3-sing) ((y2 gender) = *f)	;;SL: ;;TL: NP::NP [DET N] → [DET N] (;alignments: (x1::y1) (x2::y2) ;;x-side constraints: ((x1 agr) = *3-sing) ((x1 def) = *def) ((x2 agr) = *3-sing) ((x2 count) = +) ;;y-side constraints: ((y1 agr) = *3-sing) ((y1 def) = *def) ((y2 gender) = *f) ((y2 agr) = *3-sing) ((y2 gender) = (y1 gender))

1. Insert all constraints that appear in tr_1 and tr_2 into tr_{merged} and subsequently eliminate them from tr_1 and tr_2 .
2. Consider tr_1 and tr_2 separately. Perform all instances of operation 3 that are possible given the constraints.
3. Repeat step 1.

Figure 4 is an example of a very simple version space, seeded with only two transfer rules, a rule produced by ‘the man’ and one produced by ‘the woman’. In this case, the merged rule can be used to translate both ‘the man’ and ‘the woman’, whereas each of the seed rules can only be used to translate the NP they were produced from.

The Seeded Version Space algorithm itself is the repeated application of merging two transfer rules in a group and checking whether the merged rule is specific enough to translate correctly all those sentences that the unmerged rules could translate. If this is the case, a merge is accepted. Merging continues until no two transfer rules in the cluster can be merged any more. Note that this method is a greedy approach to generalization, without guaranteeing that the optimal (most general) transfer rule will be found. However, the method is sound with respect to allowable generalizations and it is computationally tractable.

5 Conclusions and Future Work

We presented a novel approach to learning in machine translation, a method that we hope will open MT up to a variety of languages in which little training data are available. We realize that our approach presents a large undertaking: it requires a specially adapted transfer engine, as well as a system that infers transfer rules. This paper presents the current state of our system. The focus of the future will be to scale to complex constructions and alignments. Aside from performing a baseline evaluation, we plan to refine the search through the seeded version space by defining a function that determines what merge is the best at any one step, given that there is more than one possible merge. Further, we plan to revisit the generalization operations that have been defined, so as to determine what the optimal step size of generalization should be. Also, currently no retraction is possible from overgeneralization. This issue will be addressed by adding specialization operations. The transfer engine will be extended to output partial translations if no full translation can be given. Also, work is underway to order the rule application by the complexity of the rule and the specificity of its constraints. This will be especially important as our system will be integrated into a multi-engine system, together with statistical and example-based MT methods.

References

1. Luc Bouquiaux and Jacqueline M.C. Thomas: Studying and Describing Unwritten Languages. The Summer Institute of Linguistics. (1992) 2
2. Bernard Comrie and Norval Smith: Lingua Descriptive Series: Questionnaire. In: Lingua. **42** (1977) 1–72 2
3. Hirsh Haym: Theoretical Underpinnings of Version Spaces. In: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI91). Morgan Kaufmann Publishers. (1991) 665–670 4
4. Hutchins, W. John and Somers, Harold L.: An Introduction to Machine Translation. Academic Press, London. (1992) 2
5. Tom Mitchell: Machine Learning. McGraw Hill (1996) 4
6. Mitchell, T. M.: Version Spaces: An Approach to Concept Learning. Stanford University. **December** (1978) 4
7. Katharina Probst and Ralf Brown and Jaime Carbonell and Alon Lavie and Lori Levin and Erik Peterson: Design and Implementation of Controlled Elicitation for Machine Translation of Low-density Languages. Workshop MT2010, Machine Translation Summit 2001. (2001) 2, 4
8. Katharina Probst and Lori Levin: Challenges in Automated Elicitation of a Controlled Bilingual Corpus. TMI 2002. (2002) 2
9. Trujillo, A.: Translation Engines: Techniques for Machine Translation. Springer-Verlag London Limited, London. (1999) 3

Toward a Hybrid Integrated Translation Environment

Michael Carl¹, Andy Way², and Reinhard Schärer³

¹ Laboratoire de Recherche Appliquée en Linguistique Informatique
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, Montréal, Quebec, Canada

`carl@iro.umontreal.ca`

² School of Computer Applications,
Dublin City University, Dublin 9, Ireland

`away@computing.dcu.ie`

³ Localisation Research Centre (LRC)
Department of Computer Science and Information Systems (CSIS)
University of Limerick, Limerick, Ireland

`Reinhard.Schaler@ul.ie`

Abstract. In this paper we present a model for the future use of Machine Translation (MT) and Computer Assisted Translation. In order to accommodate the future needs in middle value translations, we discuss a number of MT techniques and architectures. We anticipate a hybrid environment that integrates data- and rule-driven approaches where translations will be routed through the available translation options and consumers will receive accurate information on the quality, pricing and time implications of their translation choice.

1 A Model for the Use of MT

In this paper, we present a model for the future use of Machine Translation (MT) and Computer Assisted Translation (CAT) (cf. [22]). The model (see Figure 1) is based on the assumption that information can be categorized into three types.

At the bottom of the pyramid comes non-mission-critical information, the so-called gisting market. An example might be an article written in Japanese about Picasso on a website in Japan, of which an English speaker with no Japanese but interested in the Spanish painter wants a rough and ready translation. This is the ideal application scenario to ensure wide use of general purpose MT.

In the middle of the pyramid come large amounts of material that have to be translated accurately, where gisting is not acceptable. Examples of this type of information are technical manuals and other documentation. Most of these translations are domain-specific requiring specialized dictionaries with well defined meanings and/or specialized grammars. MT is currently being used at this level, although not widely.

At the top of the pyramid come small amounts of mission-critical or creative material to be read or referenced where accuracy and presentation are

Value of Translation

Type of Translation

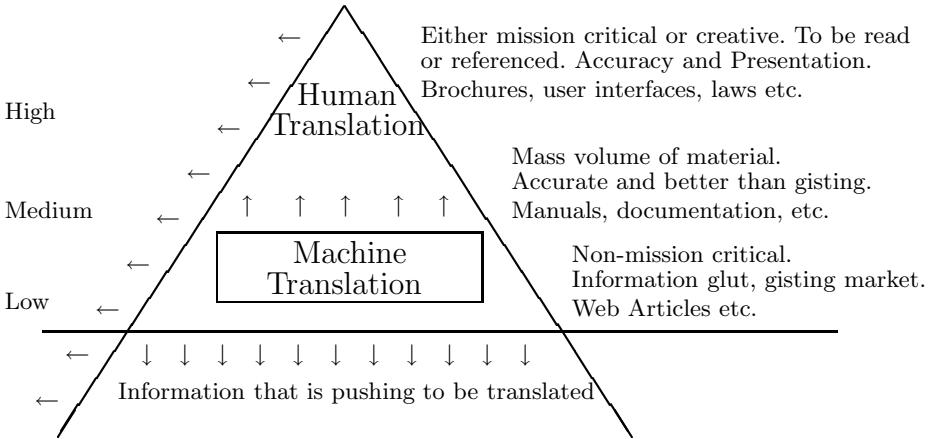


Fig. 1. A Model for the Future Use of Translation Technology

paramount. Examples of this include user interfaces, laws and creative literature.

The model presumes (1) that the shape of the pyramid is expanding in two directions and (2) that improvements in translation technology will open up new markets for developers of MT systems. The expansion of the pyramid will be driven by two factors: a growing demand for translated material because of the globalisation of the economy (horizontal expansion) and the increasing availability and accessibility of information in a variety of languages to end-users on the web (vertical expansion). At the same time, MT will push its way up the pyramid and be used for higher quality translation.

2 An Integrated Translation Environment

Translation service vendors will offer various translation facilities online, from high quality human translation to low-end, cheaper MT. In between we envisage a range of mixed options, including human-edited MT using specialized and fine-tuned lexical and semantic databases, a combination of TM and MT, and alignment and maintenance of previously translated material.

We anticipate a hybrid MT platform which integrates a number of applications, techniques and resources together. These include applications such as multilingual alignment, terminology management, induction of grammars and translation templates, and consistency checkers. These platforms will also integrate example-based, statistics-based and rule-based approaches to MT, together with a variety of other linguistic resources and corpora.

Some researchers have pondered the suitability of texts for MT. The work that we are aware of regarding translatability and MT focuses only on what texts should be sent to rule-based MT systems. One possible translatability indicator for the use of MT in general is the identification of (sets of) phenomena which are likely to cause problems for MT systems ([11], with respect to the LOGOS MT system). Based on their work with the PaTrans system, Underwood and Jongejan provide a definition of translatability:

“the notion of translatability is based on so-called ‘translatability indicators’ where the occurrence of such an indicator in the text is considered to have a negative effect on the quality of machine translation. The fewer translatability indicators, the better suited the text is to translation using MT” [26, p.363].

In an integrated translation environment, these definitions have to be widened considerably. Future translatability indicators will have to be more fine-grained and MT systems will have to be adaptable to and learn from such indicators. Translatability indicators will have to detail why a text is not (yet) suited for automatic translation so that a tool may be triggered to render the text suitable for automatic translation. That is, a hybrid integrated translation environment has to provide a means of separating translatable from non-translatable parts of a text in a more sophisticated manner than TMs currently do. For each part one has to estimate the expected quality of the translation, the effort and cost of upgrading resources and/or improving the source text in order to improve translation quality. The integrated system has to be aware of gaps in the source text which it cannot tackle and provide intelligent inference mechanisms to generate solutions for bridging these gaps.

Translations will be routed through the available translation options according to criteria such as the type of text, the value of the information to be translated, the quality requirements of the consumers, and the resources available to them. Before they select one of the translation options, consumers will receive accurate information on the quality, pricing and time implications of their choice.

3 Enhancing Medium Value Translation Quality

Despite major efforts to build new translation engines and to increase the quality of automatic translations, a major breakthrough cannot be expected in the years to come through refined technologies alone. Rather, in order to enhance the quality of MT systems and to make it suitable for medium value translations (cf. Figure 1), MT systems need to be adjusted to the domain at hand. Controlled languages and controlled translations have a crucial role to play here.

Controlled languages have been developed since the early 70’s as a compromise between the intractability of natural languages and the formal simplicity of artificial languages. Controlled languages define a writing standard for domain-specific documents. Linguistic expressions in texts are restricted to a subset of natural languages. They are characterized by simplified grammars and

style rules, a simplified and controlled vocabulary with well defined meanings, and a thesaurus of frequently occurring terms. Controlled languages are used to enhance the clarity, usability, and translatability of documents. According to Lehrndorfer and Schachtl [16, p.8], “the concept of controlled language is a mental offspring of machine translation”. A number of companies (e.g. Boeing, British Airways, Caterpillar) use controlled language in their writing environment. Nor is this trend restricted to English: Siemens use controlled German (Dokumentationsdeutsch [16]), Aérospatiale use controlled French (GIFAS Rationalized French [4]), while Scania use controlled Swedish (ScaniaSwedish [2]).

We now examine how well rule-based and data-driven MT systems may be adapted to controlled languages.

3.1 Controlled Language and Rule-Based MT

Controlled languages have been developed for restricted domains, such as technical documentation for repair, maintenance and service documents in large companies (e.g. Siemens, Scania, GM). Caterpillar’s Technical English, for instance, defines monolingual constraints on the lexicon, and constraints on the complexity of sentences. However, when using this controlled language for translation in the KANT rule-based MT (RBMT) system, it was found that: “[terms] that don’t appear to be ambiguous during superficial review turned out to have several context-specific translations in different target languages” [14].

Van der Eijk *et al.* [27, p.64] state that “an approach based on fine-tuning a general system for unrestricted texts to derive specific applications would be unnecessarily complex and expensive to develop”. Later work in METAL applications refers to there being “limits to fine-tuning big grammars to handle semi-grammatical or otherwise badly written sentences. The degree of complexity added to an already complex NLP grammar tends to lead to a deterioration of overall translation quality and (where relevant) speed” [1, p.595]. Furthermore, attempts at redesigning the Météo system, probably the biggest success story in the history of MT, to make it suitable for another domain (aviation) proved unsuccessful.

Controlled translation, therefore, involves more than just the translation of a controlled language. Passing a source language text through a controlled language tool is not sufficient for achieving high quality translation. Large general purpose MT systems cannot easily be converted to produce controlled translations. In a conventional transfer-based MT system, for instance, controlling the translation process involves controlling three processing steps: (i) the segmentation and parsing of the source text; (ii) the transfer of the source segments into the target language ; and (iii) the recombination and ordering of the target language segments according to the target language syntax. As the resources of each of these steps require independent knowledge resources, adjusting a conventional RBMT system to a new controlled language is non-trivial as domain-specific knowledge resources have to be acquired, adjusted and homogenized.

3.2 Controlled Language and Data-Driven MT

It is widely acknowledged that data-driven MT systems can overcome the ‘knowledge acquisition bottleneck’ given that available translations can be exploited. In contrast to traditional approaches, data-driven MT systems induce the knowledge required for transfer from a reference text. To date, data-driven MT technologies have yet to tackle controlled languages: they have not supported the acquisition of controlled translation knowledge, nor have they provided an appropriate environment for controlled translation.

This is extremely surprising: the quality of data-driven translation systems depends on the quality of the reference translations from which the translation knowledge was learned. The more a reference text is consistent, the better the expected quality of the translations produced by the system, while translation knowledge extracted from noisy corpora has an adverse impact on overall translation quality. The only research we are aware of here attempts to detect mistranslations [21] or omissions in translations [9], [17]. However, in the context of data-driven MT, such methods have not been used so far to eliminate noisy or mistranslated parts of the reference text, nor to enhance the quality and consistency of the extracted translation units.

Controlled Language and TM Conventional TM systems are not suitable for controlled translation. TMs are essentially bilingual databases where translation segments are stored without a built-in possibility to automatically check the consistent use of terms and their translations or the complexity of the sentences.

Within the TETRIS-IAI project [13], controlled language was fed into a TM. It was found that controlling the source language without controlling the reference material does not increase the hit-rate of the TM and thus does not increase the chance of high quality translations—from a company’s point of view, the bottom line is that the translation cost is not lowered. Methods for preparing and modifying reference texts to achieve better consistency on a terminological and syntactic level have therefore been proposed [24] and could also be a feasible way forward for TMs.

The translation process in a TM system may be distorted by two factors: the way entries are retrieved from the TM (fuzzy matching) and the contents of the TM (the chance that it contains noisy and inconsistent fragments). Where these factors co-occur, we can expect translation quality to deteriorate further.

Controlled Language and Statistics-Based MT Similarly, purely statistics-based MT (SBMT) is not an appropriate candidate for controlled translation. Owing to the size of the reference texts, one cannot usually expect consistent reference translations in SBMT. In many cases, texts from different domains are merged together to compute word translation probabilities for a language pair in various contexts. However, how words and phrases are used in different domains can differ greatly. [15] shows that the performance of a statistical MT system trained on one of the largest available bilingual texts—the Canadian Hansards—deteriorates when translating texts from a very specific domain.

Controlled Language and Example-Based MT In our view, the main potential of example-based MT (EBMT) lies in the possibility of easily generating special purpose MT systems. As in other data-driven MT systems, EBMT [8] extracts lexical and transfer knowledge from aligned texts. A number of systems (e.g. [18,25,29]) combine linguistic resources and dictionaries to support this acquisition process. Automatic and/or semi-automatic control mechanisms could be implemented at this stage to extract high quality and/or domain-specific translation equivalences [3,29].

Controlling the translation in EBMT implies the careful selection of translation examples which are similar to the input. Given that only target fragments of the retrieved examples are recombined to build the translation, controlling EBMT is reduced to controlling the retrieval of appropriate analogous examples.

The more the domain of the text to be translated is restricted, the more these restrictions are well defined and the more high quality reference translations are available, so the analogy between the new text and the retrieved examples will become more obvious. In contrast to SBMT and TM, the potential of EBMT improves as the likelihood of producing high quality translations increases as more examples are added to the system database.

[6] shows that coverage can be increased by a factor of 10 or thereabouts if templates are used, but it would be fanciful to think that this would scale up to domain-independent translation. Even if EBMT systems were augmented with large amounts of syntactic information (e.g. [18,30]), they would in all probability stop short of becoming solutions to the problems of translating general language. Nevertheless, it is our contention that EBMT systems may be able to generate controlled, domain-specific translations given a certain amount of built in linguistic knowledge and some preparation of the aligned corpus.

4 Integrating Different MT Paradigms

The various MT paradigms have different advantages and shortcomings. TMs are fed with domain-specific reference translations and are widely used as tools for CAT. TMs, however, run short of providing sufficient control mechanisms for more sophisticated translations. In contrast, RBMT systems are mostly designed for translating general purpose texts. As a consequence, they are difficult to adjust to specialized texts and consequently suffer from limited portability. Probabilistic approaches to MT are trained on huge bilingual corpora, yet portability of these systems remains low. As a compromise between the different approaches, EBMT systems have emerged as primarily data-driven systems but which may also make use of sophisticated rule-based processing devices at various stages of the translation process.

Given the different advantages and disadvantages of each MT paradigm, hybrid and multi-engine MT systems have been designed as an attempt to integrate the advantages of different systems without accumulating their shortcomings.

4.1 Multi-engine Machine Translation Systems

In order to classify these systems, a distinction can be made as to whether entire translation engines are triggered in parallel or sequentially. In a parallel multi-engine scenario, each system is fed with the source text and generates an independent translation. The translations are then collected from their output and (manually or automatically) recombined.

Parallel Multi-engine Translation Systems There are a number of projects which incorporate different MT components in parallel in a multi-engine system. Verbmobil [28] integrates the complementary strengths of various MT approaches in one framework, i.e. deep analysis, shallow dialogue act-based approach and simple TM technology. [19] shows that the performance of the integrated system outperforms each individual system. PanGloss [10] uses EBMT in conjunction with KBMT and a transfer-based engine.

While there is an element of redundancy in such approaches given that more than one engine may produce the correct translation (cf. [30]), one might also treat the various outputs as comparative evidence in favour of the best overall translation. Somers [23] observes: “what is most interesting is the extent to which the different approaches mutually confirm each other’s proposed translations”.

Sequential Multi-engine Translation Systems In this approach, two or more MT components are triggered on different sections of the same source text. The output of the different systems is then concatenated without the need for further processing. This dynamic interaction is monitored by one system — usually the most reliable amongst the available systems. The reasoning behind this approach is that if one knows the properties of the translation components involved, reliable translations can be produced by using fewer resources than in a parallel multi-engine approach.

Integration of a TM with a rule-based component is a common strategy in commercial translation. A dynamic sequential interaction between a translation memory (TRADOS) and an MT system (LOGOS) is described in [12]. In the case where only poorly matching reference translations are available in TRADOS, the input sentence is passed to LOGOS for regular translation. The user is then notified which of the systems has processed the translation, since LOGOS is less likely to produce reliable results.

A similar scenario is described in [7]. Where a TM is linked with an EBMT system, the quality of translations is likely to be higher for EBMT translation than for TM translation, where the match score of the TM falls below 80%.

4.2 Hybrid MT Systems

While in multi-engine MT systems each module has its own resources and data structures, in a hybrid MT system the same data structures are shared among different components. Some components may, therefore, modify or adjust certain

processing resources of others in order to enhance the translation candidates with respect to coverage or translation quality.

Hybrid Statistics-Based and Rule-Based MT Systems Coupling (statistical) data and RBMT leads to a hybrid integration. In some such hybrid systems, statistical data is added to the lexical resources of the RBMT system in order to adjudge different translation candidates as more or less felicitous for a given thematic context. In particular, it has been shown that statistically enriched RBMT systems can handle collocational phenomena.

[20] describes an application of statistical data during the rule-based transfer phase. Statistical data are derived by manually scoring translation variants produced by the system. Since training is based on texts belonging to one specific subject field, typical mistakes made by the system can be corrected. The probability of a transfer candidate is calculated by means of the transfer probability and the probability of the resulting target structure. As such a multiplication of probabilities requires large amounts of data in order to be effective, these approaches are applicable only to very restricted subject fields where only a few examples may suffice in order to produce reliable data. In such cases, translation quality is traded for improved coverage.

Hybrid Example-Based and Rule-Based MT Systems In a hybrid stratificational integration of example-based and rule-based techniques, some processing steps are carried out by the rule-based component while examples are used in other stages.

[18] combines rule-based analysis and generation components with example-based transfer. [5] generates translation templates for new sentences on the fly from a set of alignments. The differing sections in the source template and the input sentence are identified and translated by a rule-based noun-phrase translation system.

However, even a very large data-driven MT system is unlikely to be able to translate a completely new sentence correctly, let alone an entire new text. However, such systems are able to ‘learn’ in that new examples can be added to the system database, so that subsequent encounters with previously unknown strings will be translated successfully. In RBMT systems there is no such analogous process. That is, they do not store translation results for later reuse, so that all post-editing effort is wasted: RBMT systems handle the same input in exactly the same way in perpetuity.

A hybrid system, in contrast, will be able to learn and adapt easily to new types of text. Furthermore, such systems are based on sophisticated language models as a property of the rule-based component. Consequently, one can envisage that even if none of the individual engines can translate a given sentence correctly, the overall system may be able to do so if the engines are allowed to interact. Even if the individual components improve, the integrated system should always outperform the individual systems with respect to either the quality of the translation, the performance, or the tunability of the system.

5 Conclusion

On various occasions in recent decades, MT companies have claimed that the linguistic technology developed by them has made human translation redundant. These claims have so far not had a significant impact on the reality of translation as a profession and as a business. The one technology that has had a considerable impact on translation has been TM—it has changed the way translators work, as can be seen when examining the impact it had in the localization industry, one of the largest employers of technical translators. Ironically, TM technology works without any of the sophisticated linguistic technologies developed over decades by MT developers.

Only recently, and driven by increased activities in the area of EBMT, has the interest shown by the linguistic tools industry in research results been reciprocated by the research community. One possible reason for this development is that although EBMT as a paradigm has been described in research papers as far back as the 1980's, and although it has managed to capture the interest and enthusiasm of many researchers, it has, so far, failed to reach the level of maturity where it could be transformed from a research topic into a technology used to build a new generation of MT engines—and new approaches, technologies and applications are badly needed in MT.

If data-driven MT is to find a niche amongst the different MT paradigms, we believe it has to offer the potential to easily adapt to new domains in a more controlled manner than TMs currently do. The adaptation process differs from TM technology with respect to how and what kind of translation knowledge is stored, how it is retrieved and how it is recomposed to build a new translation. This requires sophisticated processing based on linguistic resources and/or advanced numerical processing.

In this paper we developed a model for the future use of translation technology. We anticipate an integrated hybrid translation environment which unifies a number of MT technologies, linguistic and processing resources and the actual human translator. This setting would be a valuable aid to translators, capable of generating descriptive, controlled or general translations according to the needs of users and the effort they are willing to invest.

References

1. Adriens, G., Schreurs, D.: From cogram to alcogram: toward a controlled English grammar checker. In *Coling*, Nantes, France (1992) 595–601 14
2. Almqvist, I., Sagvall Hein, A.: Defining ScaniaSwedish - a Controlled Language for Truck Maintenance. In *CLAW 96*, Leuven, Belgium (1996) 159–164 14
3. Andriamanankasina, T., Araki, K., Tochinal, K.: EBMT of POS-tagged sentences with inductive learning. In [8]. 16
4. Barthe., K.: GIFAS rationalised French: designing one controlled language to match another. In *CLAW 98*, Pittsburgh, PA. (1998) 87–102 14
5. Bond, F., Shirai, S.: A hybrid and example-based method for machine translation. In [8]. 18
6. Brown, R. D.: Example-Based Machine Translation at Carnegie Mellon University. *The ELRA Newsletter*, 5(1) (2000) 10–12 16

7. Carl, M., Hansen, S.: Linking Translation Memories with Example-Based Machine Translation. In *MT Summit VII*, Singapore (1999) 17
8. Carl, M., Way, A. (eds.): *Recent Advances in Example-Based Machine Translation*. Kluwer Academic Publishers, Boston Dordrecht London (in press) (2002) 16, 19, 19, 20, 20, 20
9. Chen, S.: Building probabilistic models for natural language. PhD thesis, Harvard University, Cambridge, MA. (1996) 15
10. Frederking, R., Nirenburg, S.: Three heads are better than one. In *Proceedings of ANLP-94*, Stuttgart, Germany (1994) 95–100 17
11. Gdaniec, C.: The LOGOS translatability index. In *Proceedings of the First Conference for Machine Translation in the Americas*, Columbia, MD. (1994) 97–105 13
12. Heyn, M.: Integrating machine translation into translation memory systems. In *EAMT Workshop Proceedings*, ISSCO, Geneva (1996) 111–123 17
13. IAI, Saarbrücken, Germany: Technologie-Transfer intelligenter Sprachtechnologie (1999) http://www.iai.uni-sb.de/tetris/tetris_home.htm 15
14. Kamprath, C., Adolphson, E., Mitamura, T., Nyberg, E.: Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English. In *CLAW '98*, Pittsburgh, PA. (1998) 14
15. Langlais, P.: Terminology to the rescue of statistical machine translation: an experiment. *To appear* (2002) 15
16. Lehrndorfer, A., Schachtl, S.: Controlled Siemens Documentary German and Top-Trans. In *TC-FORUM* (1998) 14, 14
17. Melamed, D. I.: *Empirical Methods for Exploiting Parallel Texts*. MIT Press, Cambridge, MA. (2001) 15
18. Menezes, A., Richardson, S. D.: A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In [8]. 16, 16, 18
19. Nübel, R.: End-to-End Evaluation in Verbmobil I. In *MT Summit VI*, San Diego, CA., (1997) 17
20. Rayner, M., Bouillon, P.: Hybrid transfer in an English-French spoken language translator. In *Proceedings of the IA'95*, Montpellier, France (1995) 18
21. Russell, G.: Errors of Omission in Translation. In *TMI 99*, Chester, UK (1999) 128–138 15
22. Schärer, R.: New media localisation - a linglink report for the European Commission DGXIII. Technical report, Luxembourg (1999) 11
23. Somers, H.: Review Article: Example-based Machine Translation. *Machine Translation* 14(2) (1999) 113–157 17
24. Somers, H.: The Current State of Machine Translation. In *MT Summit VI*, San Diego, CA. (1993) 115–124 15
25. Sumita, E.: An example-based machine translation system using DP-matching between word sequences. In [8]. 16
26. Underwood, N., Jongejan, B.: Translatability checker: A tool to help decide whether to use MT. In *MT Summit VIII*, Santiago de Compostela, Spain (2001) 363–368 13
27. Van der Eijk, P., Koning, M.d., Steen, G.v.d.: Controlled language correction and translation. In *CLAW 96*, Leuven, Belgium (1996) 64–73 14
28. Wahlster, W. (ed.): *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer-Verlag, Berlin Heidelberg New York (2000) 17
29. Watanabe, H.: Finding translation patterns from dependency structures. In [8]. 16, 16
30. Way, A.: LFG-DOT: A hybrid architecture for robust MT. PhD thesis, University of Essex, Colchester, UK (2001) 16, 17

Adaptive Bilingual Sentence Alignment

Thomas C. Chuang¹, GN You², and Jason S. Chang³

¹ Dept of Computer Science, Van Nung Institute of Technology
1 Van-Nung Road, Chung-Li Tao-Yuan, Taiwan, ROC
tomchuang@cc.vit.edu.tw

² Dept of Information Management
National Taichung Institute of Technology
San Ming Rd, Taichung,
Taiwan, ROC
gny@mail.ntit.edu.tw

³ Dept of Computer Science, National Tsing Hua Univ.
101, Sec. 2, Kuang Fu Rd.
Hsinchu, Taiwan, ROC
jschang@cs.nthu.edu.tw

Abstract. We present a new approach to the problem of aligning English and Chinese sentences in a bilingual corpus based on adaptive learning. While using length information alone produces surprisingly good results for aligning bilingual French and English sentences with success rates well over 95%, it does not fair as well for the alignment of English and Chinese sentences. The crux of the problem lies in greater variability of lengths and match types of the matched sentences. We propose to cope with such variability via a two-pass scheme under which model parameters can be learned from the data at hand. Experiments show that under the approach bilingual English-Chinese texts can be aligned effectively across diverse domains, genres and translation directions with accuracy rates approaching 99%.

1 Introduction

Recently, there are renewed interests in using bilingual corpus for building systems for statistical machine translation [15], computer-assisted revision of translation [7], and cross-language information retrieval [11]. It is therefore useful for the bilingual corpus to be aligned at the sentence level first. After that, further analyses such as phrase and word alignment, bilingual terminology extraction can be performed.

Pioneering work [3,6] on sentence alignment of bilingual corpus shows that length information alone is sufficient to produces surprisingly good results for aligning bilingual texts written in two closely related languages such as French-English and English-German. That degree of success can be attributed to the similarity of the two languages that reflected in two aspects. Firstly, about 90% of the aligned pairs consist of one sentence in each of the two languages (1-1 match type). Secondly, the lengths of two aligned parts are close to each other with length ratios consistently close to

unity. Therefore the performance of the aligner is insensitive to the estimates of probability of either length ratios or match types. Therefore, these probabilistic parameters are estimated in a supervised and static fashion.

However, for bilingual text from diverse language families such as Chinese-English or Japanese-English texts, it is quite a different story. Work on sentence alignment of English and Chinese texts [14] indicates that the lengths of English and Chinese texts are not as highly correlated as in French-English task, which leads to lower success rate for length based aligners. That probably can be partly attributed to the failure of properly estimating probabilities of lengths and match type.

In this paper, we examine how lengths can be best measured and length related probability best estimated to achieve high performance with a length-based sentence aligner for the English-Chinese task. We propose an adaptive approach under which length ratio of bilingual sentences is modeled by Gaussian distribution with parameters estimated on the fly based on the bilingual text at hand. We also examine the feasibility of using a probabilistic lexicon in addition to length information to obtain even more accurate alignment. Experiments were carried out to evaluate the performance of the proposed method. Experiments show that the method aligns very effectively bilingual English-Chinese texts across domains, genres and translation directions with very accuracy approaching 99%.

2 Length Based Sentence Alignment

While the length-based approach [3,6] to sentence alignment produces surprisingly good results for French, English and German texts at success rates well over 95%. We found that is not the case with bilingual texts across different language families. The performance of a length-based sentence aligner is very sensitive to the quantities of length related distribution parameters when working with the Chinese-English task. For text and translation of similar languages, e.g. English, German and French, the mean length ratio is close to unity, while for Chinese-English case, the means range from 2.96 to 4.63 with domains, genres and translation directions. The associated variance also varies considerably. It is not surprising that it is not as easy to align English and Chinese sentences in a bilingual text. However, it is not obvious that the lower performance is largely due to the inadequacy of using a static, fixed parameters related to lengths.

The match types and associated probabilistic values shown in Table 2 also indicate that the English-Chinese task is indeed much more difficult than the French-English task. For the French-English task, 90% of the alignment involve one French and one English sentence, while less than 70% of 1-1 matches for the English-Chinese task; a marked increase in complexity. The difficulty can be further accentuated by the translation style and the domain of the text which gives rise to greater variation in length and alignment match types. It can be demonstrated that if the length based method is trained using data in one domain then tested for data in another domain, the performance can suffer substantially. For instance, the accuracy rate of a length-based aligner dropped by as large as 13% when trained on bilingual *Caterpillar User's Menus* and tested on bilingual articles in a general-interest magazine, *Sinorama* [10].

Table 1. Length ratios of English-Chinese texts. The mean $\bar{x}$ and standard deviation σ of length ratios vary a great deal for bilingual text of different domains, genres and translation direction. However, the correlation R^2 of lengths for English and Chinese texts are comparable to that between German and English.

Text	$\bar{x}$	σ	R^2	Domain	Genre	Languages & Direction
Sinorama	4.63	0.76	0.92	General	Magazine	Chinese-English
Scientific American	3.36	0.40	0.87	Science	Magazine	English-Chinese
Studio Classroom	3.00	0.53	0.82	Science	Reader	English-Chinese
Harry Potter	2.96	0.49	0.93	Fiction	Novel	English-Chinese
UBS	1.06	0.22	0.99	Economy	Report	French-English-German

Table 2. Match types and distributions. For sentence alignment, the English-Chinese task is considered more difficult than the French-English task because of text and translation is asymmetrical and more even distribution of various match types.

Text	1-1	1-0	0-1	1-2	2-1	2-2	1-3	3-1
Chinese-English ¹	0.64	0.0056	0.0056	0.017	0.25	-	0.056	-
English-Chinese ²	0.67	-	0.026	0.13	0.13	-	0.026	0.026
French-English-German ³	0.89		0.0099		0.089	0.011	-	-

¹Source: Cheng and Chen, bilingual articles published in Sinorama, 1991.

²Source: Scientific American articles, www.sciam.com and www.sciam.com.tw.

³Source: Gale and Church [6], Economic Report, Union Bank of Switzerland.

Regardless of a greater variability, the lengths of aligned English and Chinese sentences are highly correlated and close to Gaussian distribution just like the French-English task. We found out that measuring length by characters (in bytes for English and double bytes for Chinese) leads to a distribution closest to Gaussian. We have generated the values of mean and the variance of paragraph length ratio between the source and target languages using several Sinorama magazine articles manually aligned. The English and Chinese length relationship measured by the characters (ASCII and BIG5 codes) shows a much high degree of correlation than previously reported for the genre of legislative debates [14] where lengths are measured by words. The results shown in Figure 1 indicate that the ratio of lengths between Chinese and English is much closer to Gaussian distribution than previously thought. With that, we can make use of the probabilistic value of the length ratio modeled as Gaussian by transforming it into a logarithmic value to be used in a standard dynamic programming scheme [6] to find a sequence of aligned sentences for a given pair of text and translation.

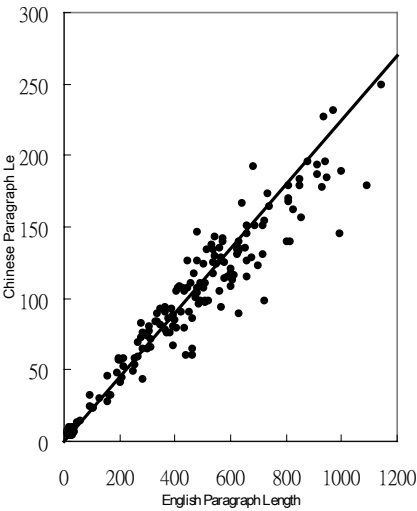


Fig. 1. The Chinese paragraph length and English paragraph length are highly correlated where the inverse of the slope represents the mean ratio between the English paragraph length to the Chinese paragraph length which is equal to 4.39. The standard deviation is equal to 0.81.

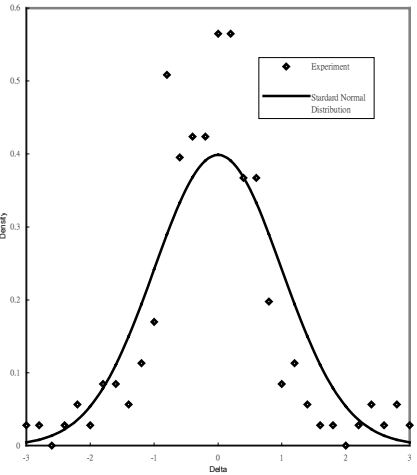


Fig. 2. The English paragraph length relative to Chinese length is close to Gaussian distribution after normalization.

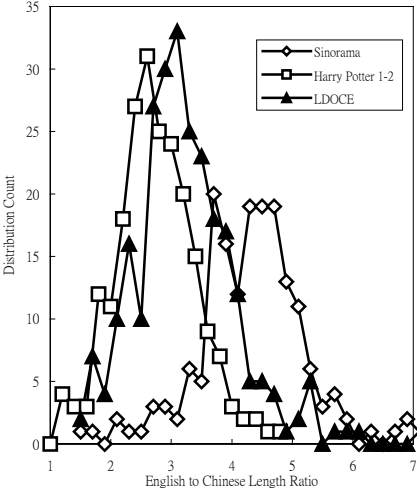


Fig. 3. The mean of length ratios vary widely across different domains and translation direction.

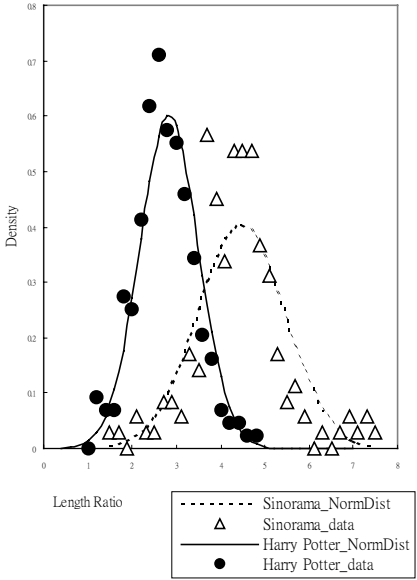


Fig. 4. Although the length ratios are consistently close to Gaussian distribution, their means can be far apart by two standard deviations.

It is therefore important to have very tight estimates of statistical parameters related to lengths and match types, not just for the bi-text *in general*, but for the bi-text *at hand*. However, in most situations it is not possible to effectively conduct supervised training for static estimation of these parameters. We make two observations that are crucial to solving this problem: Firstly, the length ratio of paragraphs are so close to that of sentences that means $\propto$ for aligned paragraphs and sentences are within 5% difference of each other for four quite different cases we have examined (See Table 3). Secondly, paragraphs are almost (over 95%) in one to one correspondence and very easy to get the alignment right. Those two observations seem to be exploitable to get tight estimates for sentence length and to improve accuracy of sentence alignment. With that in mind, we propose using the statistics of paragraph lengths to model the random distribution of sentence lengths. That can be done quite easily under a two-pass scheme where paragraphs can be aligned first before the sentences within the aligned paragraphs are aligned.

Table 3. Length ratios of English-Chinese texts. The distribution of sentence lengths and paragraph lengths is closely related. Therefore, paragraph lengths can be used to estimate mean and variance of length ratio of a pair of bilingual sentences.

Text	$\propto_{\text{paragraph}}$	$\propto_{\text{sentence}}$	$\square_{\text{paragraph}}$	$\square_{\text{sentence}}$
Sinorama	4.39	4.63	0.81	0.76
Scientific American	3.32	3.36	0.35	0.40
Studio Classroom	3.04	3.00	0.39	0.53
Harry Potter	2.78	2.96	0.52	0.49

3 Adaptive Sentence Alignment Based on Length

We will describe a self-adaptive sentence alignment program designed to cope with the problems arising from handling diverse bilingual texts. The solution hinges on adaptive, tighter estimation of length-related parameters and incorporation of word level translation probability. In this section, we deal with adaptive estimation of length related parameters. The problem of incorporating lexical information will be taken up in Section 4.

The adaptive program proceeds in two passes: paragraph alignment and sentence alignment within each aligned paragraph. The two-pass scheme allows the values of paragraph lengths to be used for sentence alignment. This simple feed-forward scheme works out with remarkable robustness and accuracy in aligning texts across dissimilar languages, domains, genres and translation directions. No *a priori* information of the domain or translation directionality of the text being handled is required for the self-adaptive calibration process. The steps of the proposed method is as follows:

1. Initial estimate of mean $\propto_0$ and variance $\square_0$ of length ratio between aligned Chinese and English sentences.

Given a small set of n bilingual sentences (E_i, C_i) , $i = 1, n$. Let X the length of English sentences relative to Chinese sentence, $X_i = |E_i| / |C_i|$ where $|E_i|$ is the length of E_i in bytes and $|C_i|$ is the length of C_i in double-bytes or bytes.

$$\alpha_0 = \frac{1}{n} \sum_{i=1}^n X_i \quad \sigma_0 = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \alpha_0)^2} \quad (1)$$

2. Set initial estimates of match types for aligned paragraphs.
3. Paragraph alignment is done by a dynamic programming algorithm centered around the calculation of $A(i, j)$, the minimum cost of aligning the first i English paragraphs and the first j Chinese paragraphs using the following recursive equation

$$D(0, 0) = 0 \quad (2)$$

$$D(i, j) = \min_{k, l} D(i - k, j - l) + m(k, l) + d(i, j, k, l) \quad (3)$$

where $m(k, l)$ is the cost of match type $k - l$ and $d(i, j, k, l)$ is length based cost of aligning k English paragraphs at the i th position and l Chinese paragraphs at the j th position.

The aligned paragraphs (P_i, Q_i) , $i = 1$ to m can be obtained through a standard back-trace process.

4. Estimation of mean α_1 and variance σ_1 of sentence length ratio using m values of length ratio, $Y_i = |P_i| / |Q_i|$, $i = 1$ to m .

$$\alpha_1 = \frac{1}{m} \sum_{i=1}^m Y_i \quad \sigma_1 = \sqrt{\frac{1}{m} \sum_{i=1}^m (Y_i - \alpha_1)^2} \quad (4)$$

5. Sentence alignment based on new values of mean and deviation. Do essentially as Step 3 except α_1 and σ_1 are used instead of α_0 and σ_0 .

In the experiment, we use a small set of 256 bilingual examples in Longman Dictionary of Contemporary Dictionary (LDOCE) to obtain the estimates of $\alpha_0 = 3.4$ and $\sigma_0 = 1.11$. The initial values for the probability of match types are derived from bilingual articles published in Sinorama magazine, 1991 given in Table 1.

4 Using Lexical Information in Addition to Length

In order to incorporate lexical information in the sentence alignment process, we take advantage of IBM Model 1 for statistical machine translation proposed by Brown et al. [2]. This is much simpler than the elaborated model proposed by Chen [5], which

takes into consideration of 1-0 and 0-1 word translation. In a nutshell, the IBM Model 1 allows us to estimate how likely a Chinese sentence C is the translation of an English sentence E by considering how likely each English word e_i in E is translated independently into c_j :

$$\Pr(C | E) = \max_A \Pr(C, A | E) = \prod_{i=1}^k \Pr(c_j | e_i) \quad (5)$$

where A is the alignment of words between E and C and $c_j = A(e_i)$.

In the work on statistical machine translation and word align [2,13], the lexical translation probability function $\Pr(c | e)$ for any English and Chinese words e and c , is estimated using a very large bilingual corpus where sentences are aligned. Here we take a simpler route by using a phrase list to take the place of bilingual sentences and derive LTP using a statistical translation model for phrases [4]. We trained a probability word translation model based on a 90,000-entry Chinese-to-English Online Dictionary [1]. C2E dictionary typically has many lexicalized Chinese entries which translate into un-lexicalized English phrases. It is at times a challenge to line up Chinese characters and English words in such a situation to obtain a word-level translation probabilistic model. Table 4 shows some instances of bilingual phrases and Viterbi alignment. Table 4 also various Chinese translations and probabilities for the word 飞行.

Table 4. Aligned words in bilingual phrases and lexical translation probability.

English Phrase E	Chinese Phrase C	A_1	A_2	e	C	$\Pr(c e)$
association football	A 式足球	A 式	足球	flight	飛行	0.648
flight eight	8 字飛行	飛行	8 字	flight	飛	0.141
delay flip-flop	D 型正反器	D 型	正反器	flight	航空	0.060
I demodulator	I 信號解調器	I 信號	解調器	flight	航	0.029
disgraceful act	不友好行動	不友好	行動	flight	分	0.004
secret ballot	不記名投票	不記名	投票	flight	分隊	0.004
bearer stock	不記名股票	不記名	股票	flight	飛航	0.004
false retrieval	不實檢索	不實	檢索	flight	飛機	0.004
used car	中古車	中古	車	□	□	□
infix operation	中序運算	中序	運算			
disregard to	不拘於	不拘	於			

Therefore, combining lengths, match type, and lexical information, the probability of aligning a pair nonempty English sentences E and English sentences C is the following

$$\Pr(C | E) = \Pr(\text{match}) \Pr(|E| / |C|) \max_A \Pr(C, A | E) \prod_{e \in E} \Pr(A(e) | e) \quad (6)$$

where A is the best alignment for the given E and $\square$ match $\square$ denotes the numbers of sentences in E and C .

Table 5. Example output of alignment bilingual article published in *Sinorama*, *Scientific American* and *Harry Potter* 1, Chaper 2

Match	English Sentences	Chinese Sentences
1-1	"We were disappointed that it wasn't a more clear-cut demonstration of an embryo that was further along," Rennie says.	他說：「我想我們對於該實驗未能清楚顯示有個更成熟的胚胎，感到失望。」
1-0	"But it was still worth doing this."	
1-1	The likelihood of intense public interest in the result as the first documented human cloning demonstration justified the decision, he explains.	但他的說詞是，大眾對於第一個複製人的紀錄可能具有的強烈興趣，讓他們下了這樣的決定。
4-1	Recently, at the invitation of Taipei City Government, Morris Chang, chairman of the powerhouse Taiwan Semiconductor Manufacturing Company (TSMC), delivered a lecture entitled "New Recruits for the 21st Century." The specification he described for modern employees included these requirements: The ability to actively participate in political and social work; an international perspective; taking pleasure in cooperative endeavor; a good general understanding of science and technology; and the ability to think independently. Finally, they should be "specialized all-rounders."	國內半導體產業龍頭台積電董事長張忠謀日前應台北市府之邀，以「二十一世紀新人才」為題發表演講，他所列出人才的「規格」包括：能積極參與政治和社會工作、具備國際視野、以合作為樂、具備科技常識、有獨立思考能力，最後要是「專精的通才」。
1-1	Finally he said slowly, "So I'll have thirty ... thirty ..."	終於他慢慢說道：「那麼我就會有三十……」
1-2	"Thirty-nine, sweetums," said Aunt Petunia.	「三十九耶，小甜心。」佩妮阿姨說。
1-0	Dudley sat down heavily and grabbed the nearest parcel.	
1-0	Uncle Vernon chuckled.	威農·德思禮姨丈咯咯輕笑。

Table 6. Adaptive learning enables a length-based aligner to improve performance dramatically, while adding lexical information pushes the accuracy rates toward 99%.

Text\Method	Static	Static+Lex	Adaptive	Adaptive+Lex
Sinorama	91%	93%	96%	98%
Scientific American	97%	97%	98%	99%
Studio Classroom	99%	99%	99%	99%
Harry Potter	98%	98%	98%	99%

The probabilistic value can be transformed into a logarithmic value to be used in the dynamic programming step in the adaptive algorithm described in Section 4 to find a sequence of non-crossing aligned sentences for a given pair of bilingual paragraphs.

5 Experiments and Evaluation

The proposed methods described in Sections 3 and 4 were tested on bilingual texts with different domains, genres and translation directions. We tested on children literature, scientific reportage, and articles from general-interest magazine, which include translations from either translation. To evaluate the results of sentence alignment, we need human judgment on whether a pair of sentences is correctly aligned which is quite straightforward, in most cases. Therefore, we only ask a judge to rate the result as `correct` and `incorrect`.

The aligner's innovative way of handling diversity in length statistics and incorporation of lexical information enable it to align very effectively bilingual texts across diverse languages, domains, genres and translation directions with accuracy rates approaching 99%, which has been achieved only for French-English task. The program aligns bilingual texts from *Harry Potter*, *Scientific American*, and *bilingual Sinorama Magazine* with equal ease and consistently high accuracy. The aligner is also good at finding small omissions; it spotted a short sentence omitted in the translation of *Harry Potter*, Vol. 1, Chapter 2, published in Taiwan:

Dudley sat down heavily and grabbed the nearest parcel.

6 Conclusion

The proposed methods show some interesting observations and results. First, for the Chinese and English task, the character length distribution is indeed close to Gaussian distribution and length based sentence alignment scheme can be applied to the Chinese/English translation.

Secondly, unlike the translation of similar languages, e.g. English, German and French where the mean paragraph length is close to unity, the value can range from 2.96 to 4.63 for Chinese to English translation. It is therefore empirical to train the length-based model using the data that is in the same translation direction. It is not always possible in most situations to use static training data from the same domain as the input data, but that seems to be not as significant as the factor of directionality. A self-adaptive calibration method is developed so that optimum length distribution mean and variance can be determined during the alignment process without prior knowledge of the translation direction or domain information.

Last but not the least, the combination of length and lexical information generates remarkable results. The methods described in this paper represent an innovative way to automatically optimize the alignment of the translation of languages in very different languages. It is especially effective in the case of translations produced in either direction that may involve cross-domain translation with omission and/or insertion of texts.

Acknowledgements

We would like to thank Wang Ying of *Sinorama Magazine* and Ivan Tsai of Yuan Liu Publishing and *Scientific American*, Taiwan for providing bilingual corpora for the experiments. The research is partially funded by National Science Council, under contract NSC 90-2411-H-007-033-MC.

References

1. Behavior Design Co.: The BDC Chinese-English Electronic Dictionary (Version 2.0), Taiwan (1992).
2. Brown, P.F., Della Pietra, S., Della Pietra, V., Mercer, R.L.: The Mathematic of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19:2 (1994) 263-311.
3. Brown, PF, Jennifer C. Lai, and Robert L. Mercer: Aligning Sentences in Parallel Corpora, In Proc. of the 29th Annual Meeting of the ACL (1991) 169-176.
4. Chang, J.S., Yu, D. and Lee, C.J.: Statistical Translation Model for Phrases, *Computational Linguistic and Chinese Language Processing*, 6:2 (2001) 43-64 (in Chinese).
5. Chen, S.F.: Aligning Sentences in Bilingual Corpora Using Lexical Information, In Proc. of 30th Annual Meeting of ACL (1993) 9-16.
6. Gale, W.A. and Church, K.W.: A program for aligning sentences in bilingual corpora, In Proc. of the 29th Annual Meeting of the ACL (1991) 177-184.
7. Jutras, J-M.: An Automatic Reviser: The TransCheck System, In Proc. of Applied Natural Language Processing (2000) 127-134.
8. Kay, M. and Ršcheisen, M: Text-Trans lation Alignment, *Computational Linguistics* 19:1 (1994) 121-142.
9. Ker, S.J. and Chang J.S.: A Class-base Approach to Word Alignment, *Computational Linguistics*, 23:2 (1997) 313-343.
10. Kueng, T.L. and Su, K.Y.: A Robust Cross-Domain Bilingual Sentence Alignment Model, In Proceedings of the 19th International Conference on Computational Linguistics (2002).
11. Kwok, K.L.: *NTCIR-2 Chinese, Cross-Language Retrieval Experiments Using PIRCS*. In Proceedings of the Second NTCIR Workshop Meeting, National Institute of Informatics, Japan (2001) 14-20.
12. Longman Group.: Longman English-Chinese Dictionary of Contemporary English, Published by Longman Group (Far East) Ltd., Hong Kong (1992).
13. Melamed, I.D.: Bitext Maps and Alignment via Pattern Recognition, *Computational Linguistics* 25:1 (1999) 107-130.
14. Wu, D.K.: Aligning a Parallel English-Chinese Corpus Statistically with Lexical Criteria, In Proc. of the 31st Annual Meeting of the Association for Computational Linguistics (1994) 80-87.
15. Yamada, K, and Knight, K.: A Syntax-based Approach to Statistical Machine Translation. Proc. of the Conference of the Association for Computational Linguistics (2001) 523-530.

DUSter: A Method for Unraveling Cross-Language Divergences for Statistical Word-Level Alignment

Bonnie J. Dorr, Lisa Pearl, Rebecca Hwa, and Nizar Habash

Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20740
{bonnie,llsp,hwa,habash}@umiacs.umd.edu
<http://umiacs.umd.edu/labs/CLIP>

Abstract. The frequent occurrence of *divergences*—structural differences between languages—presents a great challenge for statistical word-level alignment. In this paper, we introduce DUSter, a method for systematically identifying common divergence types and transforming an English sentence structure to bear a closer resemblance to that of another language. Our ultimate goal is to enable more accurate alignment and projection of dependency trees in another language without requiring any training on dependency-tree data in that language. We present an empirical analysis comparing the complexities of performing word-level alignments with and without divergence handling. Our results suggest that our approach facilitates word-level alignment, particularly for sentence pairs containing divergences.

1 Introduction

Word-level alignments of bilingual text (bitexts) are not only an integral part of statistical machine translation models, but also useful for lexical acquisition, treebank construction, and part-of-speech tagging [26]. The frequent occurrence of *divergences*—structural differences between languages—presents a great challenge to the alignment task.¹ In this paper, we introduce DUSter (Divergence Unraveling for Statistical Translation), a method for systematically identifying common divergence types and transforming the structure of an English sentence to bear a closer resemblance to that of another language.² Our ultimate goal is to enable more accurate projection of dependency trees for non-English languages without requiring any training on dependency-tree data in those languages. (For ease of readability, we will henceforth refer to *non-English* as *foreign*.) The bitext is parsed on the English side only. Thus, the projected trees in the foreign language may serve as input for training parsers in a new language.

¹ The term *divergence* refers only to differences that are relevant to predicate-argument structure, i.e., we exclude constituent re-orderings such as noun-adjective swapping which occurs between Spanish and English. See [25] for an approach that involves syntactic reorderings of this type.

² See <http://www.umiacs.umd.edu/labs/CLIP/DUSter.html> for more details.

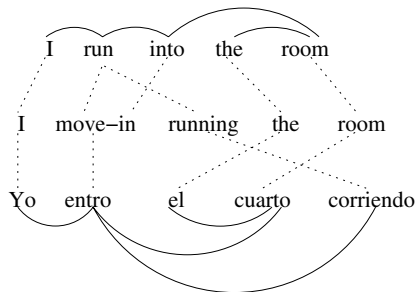


Fig. 1. Idealized Version of Transformation/Alignment/Projection

A divergence occurs when the underlying concepts or gist of a sentence is distributed over different words for different languages. For example, the notion of running into the room is expressed as *run into the room* in English and *move-in the room running* (*entrar el cuarto corriendo*) in Spanish. While seemingly transparent for human readers, this throws statistical aligners for a serious loop. Far from being a rare occurrence, our preliminary investigations revealed that divergences occurred in approximately 1 out of every 3 sentences.³ Thus, finding a way to deal effectively with these divergences and repair them would be a massive advance for bilingual alignment.

The following three ideas motivate the development of automatic “divergence correction” techniques:

1. Every language pair has translation divergences that are easy to recognize.
2. Knowing what they are and how to accommodate them provides the basis for refined word-level alignment.
3. Refined word-level alignment results in improved projection of structural information from English to another language.

This paper elaborates primarily on points 1 and 2. Our ultimate goal is to set these in the context of 3, i.e., for training foreign-language parsers to be used in statistical machine translation.

DUSTer transforms English into a pseudo-English form (which we call E') that more closely matches the physical form of the foreign language, e.g., “run into the room” is transformed to a form that roughly corresponds to “move-in the room running” if the foreign language is Spanish. This rewriting of the English sentence increases the likelihood of one-to-one correspondences which, in turn, facilitates our statistical alignment process. In theory, our rewriting approach applies to all divergence types. Thus, given a corpus, divergences are identified, rewritten, and then run through the statistical aligner of choice.

The idealized version of our transformation/alignment/projection approach is illustrated for an English-Spanish pair in Figure 1. Dependencies between

³ This analysis was done using automatic detection techniques—followed by human confirmation—on a sample size of 19K sentences from the TREC El Norte Newspaper (Spanish) Corpus, LDC catalog no LDC2000T51, ISBN 1-58563-177-9, 2000.

English words (**E**) are represented by the curves above the words—these are produced by the Minipar system [15,16]. Alignments are indicated by dotted lines. The dependency trees are transformed into new trees associated with E' , e.g., *run* and *into* in **E** are reconfigured in E' so that the sentence in E' has a one-to-one correspondence with the sentence of the foreign language **F**. The final step—outside the scope of this paper—is to induce foreign-language dependency trees automatically using statistical alignment of the E' words with those of the foreign-language sentence (e.g., using Giza++ [1,20]).

The next section sets this work in the context of related work on alignment and projection of structural information between languages. Section 3 describes the range of divergence types covered in this work—and analyzes the frequency of their occurrence in corpora (with examples in Spanish and Arabic). Section 4 describes an experiment that reveals the benefits of injecting linguistic knowledge into the alignment process. We present an empirical analysis comparing the complexities of performing word-level alignments with and without divergence handling. We conclude that annotators agree with each other more consistently when performing word-level alignments on bitext with divergence handling.

2 Related Work

Recently, researchers have extended traditional statistical machine translation (MT) models [4,5] to include the syntactic structures of the languages [2,3,23]. These statistical transfer systems appear to be similar in nature to what we are proposing—projecting from English to a foreign-language tree—but both the method of generation and the goal behind these approaches are different from ours. In these alternative approaches, parses are generated simultaneously for both sides whereas, in our approach, we assume we only have access to the English parses and then we automatically produce dependency trees in another language *without training*.⁴ From these noisy foreign-language dependency trees, we then induce a parser for translation between the foreign language and English. The foreign-language parse is a necessary input to a generation-heavy decoder [8], which produces English translations from foreign-language dependency trees.

It has been shown that MT models are significantly improved when trained on syntactically *annotated* data [25]. However, the cost of human labor in producing annotated treebanks is often prohibitive, thus rendering manual construction of such data for new languages infeasible. Some researchers have developed techniques for fast acquisition of hand-annotated Treebanks [7]. Others have developed machine learning techniques for inducing parsers [10,11], but these require extensive collections of complex translation pairs for broadscale MT.

Because divergences generally require a combination of lexical and structural manipulations, they are handled traditionally through the use of transfer rules

⁴ It is important to note that rewriting the English structure as a structure in the foreign language is not intended to be an MT transfer process unto itself, but rather it is a first step in constructing a (noisy) foreign-language treebank for training a new parser for MT.

Table 1. Examples of True English, E' , and Foreign Equivalent

Type	English	E'	Foreign Equivalent
Light Verb	fear	have fear	tiene miedo
	try	put to trying	poner a prueba
	make any cuttings	wound	تَجَرَّحُوا
	our hand is high	our hand heightened	يَدُنَا قَدْ عَظُمَتْ
	he is not here	he be-not here	إنه ليس هنا
Manner	teaches	walks teaching	anda enseñando
	is spent	self goes spending	se va gastando
	he sent his brothers away	he dismissed his brothers	صَرَفَ إِخْوَتَهُ
	spake good of	speak-good about	يُثْنِي عَلَى
	he turned again	he returned	رَجَعَ
Structural	after six years	after of six years	después de seis años
	and because of that in other parts	and for that in other parts	y por ello en otras partes
	I forsake thee	I-forsake about-you	اتَّخَلَّى عَنْكَ
	we found water	we-found on-water	عَثَرْنَا عَلَى
Categorical	I am jealous	I have jealousy	tengo celos
	I require of you	I require-of you	te pido
	(he) shall estimate	according-to (his)-estimate	حَسَبَ تَقْدِيرِ
	(how long shall) the land mourn	(stays) the-land mourning	الْأَرْضُ نَائِحَةٌ
	he went to	his-return to	عُودَتُهُ إِلَى
Head-Swapping ⁵	walked out	move-out walking	salir caminando
Thematic	I am pained	me pain they	me duelen
	He loves it	to him be-loved it	le gusta
	it was on him	he-wears-it	كان يرتديه

[9,13]. Unfortunately, automatic extraction of such rules relies crucially the availability of scarce resources such as large, aligned, and parsed, bilingual corpora [14,18,19,22].

Our approach requires parsing on only the English side of the aligned bilingual corpora—the foreign language need not be parsed. We detect and handle divergences using linguistically motivated techniques to transform the English lexical and syntactic representation to match the physical form of the foreign language more closely—thus improving alignment. The ultimate goal is to bring about more accurate dependency-tree projection from English into the foreign language, thus producing a significantly noise-reduced dependency treebank for training foreign-language parsers.

3 Frequency of Divergences in Large Corpora

We investigated divergences in Arabic and Spanish corpora to determine how often such cases arise.⁶ Our investigation revealed that there are six divergence

⁵ Although cases of Head Swapping arise in Arabic, we did not find any such cases in the small sample of sentences that we human checked in the Arabic Bible.

⁶ For Spanish, we used TREC Spanish Data; for Arabic, we used an electronic, version of the Bible written in Modern Standard Arabic.

Table 2. Common Search Terms for Divergence Detection

Spanish	Arabic
hacer (do)	ليس (be-not)
dar (give)	عبر (go-across)
tomar (take)	احسن (do-good)
tener (have)	اكتفى (make-do)
poner (put)	اخرج (take-out)
ir + X-progressive (go X-ing)	رجع (come-again)
andar + X-progressive (walk X-ing)	غرب (go-west)
salir + X-progressive (leave X-ing)	اسرع (do-quickly)
pasar + X-progressive (pass X-ing)	جرح (make-cuttings)
entrar + X-progressive (enter X-ing)	عظم (become-high/great)
bajar + X-progressive (go-down X-ing)	صرف (send-away)
irse + X-progressive (leave X-ing)	ارتعب (be-afraid)
soler (usually)	علي + اثني (speak-good + about = laud)
gustar (like)	عن + بحث (search + for = seek)
bastar (be enough)	ب + اوصى (command + with = command)
disgustar (dislike)	عن + تخلى (abandon + of = forsake)
quedar (be left over)	علي + عثر (find + on = find)
doler (hurt)	الى + عودة (return _{verb} to)
encantar (be enchanted by)	
importar (be important)	
interesar (interest)	
faltar (be lacking)	
molestar (be bothered by)	
fascinar (be fascinated by)	

types of interest. Table 1 shows examples of each type from our corpora, along with examples of sentences that were aligned with the foreign language sentences in our experiment (including both English and E').

Space limitations preclude a detailed analysis of each divergence type, but see [6] for more information. In a nutshell, Light Verb divergence involves the translation of a single verb to a combination of a “light” verb (carrying little or no specific meaning in its own right) and some other meaning unit (perhaps a noun) to convey the appropriate meaning. Manner divergence involves translating a single manner verb (e.g., *run*) as a light verb of motion and a manner-indicating content word. Structural divergence involves the realization of incorporated arguments such as subject and object as obliques (i.e. headed by a preposition in a PP). Categorical divergence involves a translation that uses different parts of speech. Head swapping involves the demotion of the head verb and the promotion of one of its modifiers to head position. Finally, a thematic divergence occurs when the verb’s arguments switch thematic roles from one language to another.

In order to conduct this investigation, we developed a set of hand-crafted regular expressions for detecting divergent sentences in Arabic and Spanish corpora (see Table 2).⁷ The Arabic regular expressions were derived by examining a small set of sentences (50), a process which took approximately 20 person-

⁷ The regular expressions are overgenerative in the current version, i.e., the system detects more seemingly divergent sentences than actually exist. Thus, we require

Table 3. Divergence Statistics

Language	Detected Divergences	Human Confirmed	Sample Size (sentences)	Corpus Size (sentences)
Spanish	11.1%	10.5%	19K	150K
Arabic	31.9%	12.4%	1K	28K

hours. The Spanish expressions were derived by a different process—involving a more general analysis of the behavior of the language—taking approximately 2 person-months. We want to emphasize that these regular expressions are *not* a sophisticated divergence detection technique. However, they do establish, at the very least, a conservative lower bound for how often divergences occur since the regular expressions pull out select cases of the different divergence types.

In our investigation, we applied the Spanish and Arabic regular expressions to a sample size of 19K Spanish sentences from TREC and 1K Arabic sentences from the Arabic Bible. Each automatically detected divergence was subsequently human verified and categorized into a particular divergence category. Table 3 indicates the percentage of cases we detected automatically and also the percentage of cases that were confirmed (by humans) to be actual cases of divergence.

It is important to note that these numbers reflect the techniques used to calculate them. The Arabic regular expressions were constructed more compactly than the Spanish ones in order to increase the number of verb forms that could be caught with a single expression. For example, a regular expression for a transitive verb includes the perfect and imperfect forms of the verb with various prefixes for conjugation, aspect, and tense and suffixes for pronominal direct objects. Because the Spanish regular expressions were derived through a more general language analysis, the precision is higher in Spanish than it is in Arabic. Human inspection confirmed approximately 1995 Spanish sentences out of the 2109 that were automatically detected (95% accuracy), whereas 124 sentences were confirmed in the 319 detected Arabic divergences (39% accuracy).

On the other hand, the more constrained Spanish expressions appear to give rise to a lower recall. In fact, an independent study with more relaxed regular expressions on the same 19K Spanish sentences resulted in the automatic detection of divergences in 18K sentences (95% of the corpus), 6.8K of which were confirmed by humans to be correct (35% of the corpus). Future work will involve repeated constraint adjustments on the regular expressions to determine the best balance between precision and recall for divergence detection; we believe the Arabic expressions fall somewhere in between the two sets of Spanish expressions (which are conjectured to be at the two extremes of constraint relaxation—very tight in the case above and very loose in our independent study).

human post-checking to eliminate erroneous cases. However, a more constrained automated version is currently under development—to be released in 2002—that requires no human checking. We aim to verify the accuracy of the automated version using some of the test data developed for this earlier version.

4 Experiment: Impact of Divergence Correction on Alignment

To evaluate our hypothesis that transformations of divergent cases can facilitate the word-level alignment process, we have conducted human alignment studies for two different pairs of languages: English-Spanish and English-Arabic. We have chosen these two pairings to test the generality of the divergence transformation principle.

Our experiment involves four steps:

- i. Identify canonical transformations for each of the six divergence categories.
- ii. Categorize English sentences into one of the 6 divergence categories (or “none”) based on the foreign language.
- iii. Apply the appropriate transformations to each divergence-categorized English sentence, renaming it E' .
- iv. For each language:
 - Have two humans align the true English sentence and the foreign-language sentence.
 - Have two different humans align the rewritten E' sentence and the foreign-language sentence.
 - Compare inter-annotator agreement between the first and second sets.

We accommodate divergence categories by rewriting dependency trees produced by the Minipar system so that they are parallel to what would be the equivalent foreign-language dependency tree. Simultaneously, we automatically rewrite the English sentence as E' . For example, in the English-Spanish case of *John kicked Mary*, our system rewrites the English dependency tree as a new dependency tree corresponding to the sentence *John gave kicks to Mary*. The resulting E' (which would be seen by the human aligner in our experiment) is: ‘John LightVB kick Prep Mary’.

The canonical transformation rules that map an English sentence and dependency tree to E' (and its associated dependency tree) are shown in Table 4. These rules fall into two categories, those that facilitate the task of alignment *and* enable more accurate projection of dependency trees (light verb, manner, and structural)—and those that *only* enable more accurate projection of dependency trees with minimal or no change to alignment accuracy (categorical, head-swapping, and thematic). This paper focuses on the first of these two categories.⁸ In this category, there are of two types of rules: “expansion rules,” applicable when the foreign language sentence is verbose relative to the English one, and “contraction rules,” applicable when the foreign language sentence is terse relative to English.⁹

⁸ The impact of our approach dependency-tree projection will be reported elsewhere and is related to ongoing work by [12].

⁹ Our empirical results show that the expansion rules apply more frequently to Spanish than to Arabic, whereas the reverse is true of the contraction rules. This is not surprising because, in general, Spanish is verbose relative to English, where as Arabic tends to be more terse. Such differences in verbosity are well documented in the

Table 4. Transformation Rules between E and E' **I. Rules Impacting Alignment and Projection****(1) Light Verb**Expansion: [Arg1 [V]] $\rightarrow$ [Arg1 [LightVB] Arg2(V)]Ex: $\square$ I fear $\square \rightarrow \square$ I have fear $\square$ Contraction: [Arg1 [LightVB] Arg2] $\rightarrow$ [Arg1 [V(Arg2)]]Ex: $\square$ our hand is high $\square \rightarrow \square$ our hand heightened $\square$ **(2) Manner**Expansion: [Arg1 [V]] $\rightarrow$ [Arg1 [MotionV] Modifier(V)]Ex: $\square$ I teach $\square \rightarrow \square$ I walk teaching $\square$

Contraction:

[Arg1 [MotionV] Modifier] $\rightarrow$ [Arg1 [V-Modifier]]Ex: $\square$ he turns again $\square \rightarrow \square$ He returns $\square$ **(3) Structural**Expansion: [Arg1 [V] Arg2] $\rightarrow$ [Arg1 [V] Oblique Arg2]Ex: $\square$ I forsake thee $\square \rightarrow \square$ I forsake of thee $\square$ Contraction: [Arg1 [V] Oblique Arg2] $\rightarrow$ [Arg1 [V] Arg2]Ex: $\square$ I search for him $\square \rightarrow \square$ I search him $\square$ **II. Rules Impacting Projection Only****(4) Categorical:**[Arg1 [V] Adj(Arg2)] $\rightarrow$ [Arg1 [V] N(Arg2)]Ex: $\square$ I am jealous $\square \rightarrow \square$ I have jealousy $\square$ **(5) Head-Swapping:**

[Arg1 [MotionV] Modifier(Direction)]

 $\rightarrow$ [Arg1 [V-Direction] Modifier(Motion)]Ex: $\square$ I run in $\square \rightarrow \square$ I enter running $\square$ **(6) Thematic:**[Arg1 [V] Arg2] $\rightarrow$ [Arg2 [V] Arg1]Ex: $\square$ He wears it $\square \rightarrow \square$ It is-on him $\square$

For each language pair, four fluently bilingual human subjects were asked to perform word-level alignments on the same set of sentences selected from the Bible. They were all provided the same instructions and software, similar to the methodology and system described by [17]. Two of the four subjects were given the original English and foreign language sentences; they served as the control for the experiment. The sentence given to the other two consisted of the original foreign language sentences paired with altered English (denoted as E') resulting from divergence transformations described above. We compare the inter-annotator agreement rates and other relevant statistics between the two sets of human subjects. If the divergence transformations had successfully modified English structures to match those of the foreign language, we would expect the inter-annotator agreement rate between the subjects aligning the E' set to be higher than the control set. We would also expect that the E' set would have fewer unaligned and multiply-aligned words.

In the case of English-Spanish, the subjects were presented with 150 sentence pairs from the English and Spanish Bibles. The sentence selection procedure is similar to the divergence detection process described in the previous section. These sentences were first selected as potential divergences, using the hand-crafted regular expressions referred to in Section 3; they were subsequently verified by the experimenter as belonging to a particular divergence type. Out of the 150 sentence pairs, 97 were verified to have contained divergences; moreover, 75 of these 97 contain expansion/contraction divergences (i.e., divergence transformations that result in altered surface words). The average length of the English sentences was 25.6 words; the average length of the Spanish sentences was 24.7 words. Of the four human subjects, two were native Spanish speakers, and two were native English speakers majoring in Spanish literature. The backgrounds of the four human subjects are summarized in Table 5.

literature. For example, according to [21], human translators often make changes to produce Spanish sentences that are longer than the original English sentence—or they generate sentences of the same length but *reduce* the amount of information conveyed in the original English.

Table 5. Summary of the Backgrounds of the English-Spanish Subjects

	data set	native-tongue	linguistic knowledge?	ease with computers
Subject 1	control	Spanish	yes	high
Subject 2	control	Spanish	no	low
Subject 3	divergence	English	no	high
Subject 4	divergence	English	no	low

Table 6. Summary of Backgrounds of English-Arabic Subjects

	data set	native-tongue	linguistic knowledge?	ease with computers
Subject 1	control	Arabic	yes	high
Subject 2	control	Arabic	no	high
Subject 3	divergence	Arabic	no	high
Subject 4	divergence	Arabic	no	high

Table 7. Results of Two Experiments on All Sentence Pairs¹⁰

	# of sentences	F-score	% of unaligned words	Avg. alignments per word
E-S	150	80.2	17.2	1.35
E^d -S	150	82.9	14.0	1.16
E-A	50	69.7	38.5	1.48
E^d -A	50	75.1	11.9	1.72

Table 8. Results for Subset Containing only Divergent Sentences

	# of sentences	F-score	% of unaligned words	Avg. alignments per word
E-S	97	81.0	17.3	1.35
E^d -S	97	83.8	13.8	1.16
E-A	50	69.7	38.5	1.48
E^d -A	50	75.1	11.9	1.72

In the case of English-Arabic, the subjects were presented with 50 sentence pairs from the English and Arabic Bibles. While the total number of sentences was smaller than the previous experiment, every sentence pair was verified to contain at least one divergence. Of these 50 divergent sentence pairs, 36 of them contained expansion/contraction divergences. The average English sentence length was 30.5 words, and the average Arabic sentence length was 17.4 words. The backgrounds of the four human subjects are summarized in Table 6.

Inter-annotator agreement rate is quantified for each pair of subjects who viewed the same set of data. We hold one subject’s alignments as the “ideal” and compute the precision and recall figures for the other subject based on how many alignment links were made by both people. The averaged precision and recall figures (F-scores)¹¹ for the two experiments and other relevant statistics are summarized in Table 7. In both experiments, the inter-annotator agreement is higher for the bitext in which the divergent portions of the English sentences have been transformed. For the English-Spanish experiment, the agreement rate increased from 80.2% to 82.9% (error reduction of 13.6%). Using the pair-wise

¹⁰ In computing the average number of alignments per word, we do not include unaligned words.

¹¹ $F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Table 9. Results for Subset Containing only Sentence Pairs with Expansion/Contraction Divergences

	# of sentences	F-score	% of unaligned words	Avg. alignments per word
E-S	75	82.2	17.3	1.34
E^d -S	75	84.6	13.9	1.14
E-A	36	69.1	38.3	1.48
E^d -A	36	75.7	11.5	1.67

t-test, we find that the higher agreement rate is statistically significant with 95% confidence. For the English-Arabic experiment, the agreement rate increased from 69.7% to 75.1% (error reduction of 17.8%); this higher agreement rate is statistically significant with a confidence rate of 90%.

We also performed data analyses on two subsets of the full study. First, we focused on sentence pairs that were verified to contain divergences; the results are reported in Table 8. They were not significantly different from the complete set. We then considered a smaller subset of sentence pairs containing only expansion/contraction divergences whose transformations altered the surface words as well as the syntactic structures; the results are reported in Table 9. In this case, the higher agreement-rate for the English'-Spanish annotators is statistically significant with 90% confidence; the higher agreement-rate for the English'-Arabic annotators is statistically significant with 95% confidence.

Additional statistics also support our hypothesis that transforming divergent English sentences facilitates word-level alignment by reducing the number of unaligned and multiply-aligned words. In the English-Spanish experiment, both the appearances of unaligned words and multiply-aligned words decreased when aligning to the modified English sentences. The percentage of unaligned words decreased from 17% to 14% (18% fewer unaligned words), and the average number of links to a word is lowered from 1.35 to 1.16.¹² In the English-Arabic experiment, the number of unaligned words is significantly smaller when aligning Arabic sentences to the modified English sentences; however, on average multiple-alignment increased. This may be due to the big difference in sentence lengths (English sentences are typically twice as long as the Arabic ones); thus it is not surprising that the average number of alignments per word would be closer to two when most of the words are aligned. The reason for the lower number in the unmodified English case might be that the subjects only aligned words that had clear translations.

5 Conclusion and Future Work

In this paper, we examined the frequency of occurrence of six divergence types in English-Spanish and English-Arabic. By examining bitext corpora, we have established conservative lower-bounds, estimating that these divergences occur

¹² The relatively high overall percentage of unaligned words is due to the fact that the subjects did not align punctuation marks.

at least 10% of the time. A realistic sampling indicates that the percentage is actually significantly higher, approximately 35% in Spanish.

We have shown that divergence cases can be systematically handled by transforming the syntactic structures of the English sentences to bear a closer resemblance to those of the foreign language, using a small set of templates. The validity of the divergence handling has been verified through two word-level alignment experiments. In both cases, the human subjects consistently had higher agreement rate with each other on the task of performing word-level alignment when divergent English phrases were transformed.

The results of this work suggest several future research directions. First, we are actively working on automating the process of divergence detection and classification, with the goal of replacing our “common search terms” in Table 2 with automatic detection routines based on parameterization of the transformation rules in Table 4.¹³ Once the process has been automated, we will be able to perform large-scaled experiments to study the effect of divergence handling on statistical word-alignment models.

Second, while we have focused on the effect of divergence handling on the word-alignment process in this work, we also need to evaluate the effect of divergence handling on the foreign parse trees. Our latest experiments involve projection of English-Chinese experiments; we will evaluate whether our transformation rules on the English structures result in better projected Chinese dependency structures by evaluating against Chinese Treebank data [24].

Finally, we plan to compare our approach with that of [12] in creating foreign language treebanks from projected English syntactic structures. Both approaches apply techniques to improve the accuracy of projected dependency trees, but ours occurs *prior* to statistical alignment, making corrections relevant to general divergence classes—whereas the latter occurs *after* statistical alignment, making corrections relevant to syntactic constraints of the foreign language. We will evaluate different orderings of the two different correction types to determine which ordering is most appropriate for optimal projection of foreign-language dependency trees.

Acknowledgments

This work has been supported, in part, by ONR MURI Contract FCPO.810548265 and Mitre Contract 010418-7712. We are grateful for the assistance of our Span-

¹³ For example, we will make use of lexical parameters such as LightVB, MotionV, and Oblique for our Type I rules. We already adopt the LightVB parameter in our current scheme—the current setting is {do, give, take, put, have} in English and {hacer, dar, tomar, tener, put} in Spanish. Settings for MotionV and Obliques are also available for English—and preliminary settings have been assigned in Spanish and Arabic. Three additional parameters will be used for Type II rules—Direction, Swap, and CatVar—the latter one associated with categorial variation which will be semi-automatically acquired using resources developed in the categorial variation work of [8]. All settings are small enough to be constructed in 1 person-day by a native speaker of the foreign language.

ish aligners, Irma Amenero, Emily Ashcraft, Allison Bigelow, and Clara Cabezas; and also our Arabic aligners, Moustafa Al-Bassyouni, Eiman Elnahrawy, Tamer Nadeem, and Musa Nasir.

References

1. Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, I.D., Och, F.J., Purdy, D., Smith, N.A., Yarowsky, D.: Statistical machine translation: Final report. In: *Proceedings of the Summer Workshop on Language Engineering*. John Hopkins University Center for Language and Speech Processing (1999) 33
2. Alshawhi, H., Douglas, S.: Learning Dependency Transduction Models from Unannotated Examples. *Philosophical Transactions, Series A: Mathematical, Physical and Engineering Sciences* (2000) 33
3. Alshawhi, H., Bangalore, S., Douglas, S.: Learning Dependency Translation Models as Collections of Finite State Head Transducers. *Computational Linguistics*. Vol. 26 (2000) 33
4. Brown, P.F., Cocke, J., Della-Pietra, S., Della-Pietra, V.J., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A Statistical Approach to Machine Translation. *Computational Linguistics*. Vol. 16(2) (1990) 79-85 33
5. Brown, P.F., Della-Pietra, S.A., Della-Pietra, V.J., Mercer, R.L.: The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*. (1993) 33
6. Dorr, B.J., Pearl, L., Hwa, R., Habash, N.: Improved Word-Level Alignment: Injecting Knowledge about MT Divergences. University of Maryland Technical Report LAMP-TR-082, CS-TR-4333, UMIACS-TR-2002-15 College Park, MD. (2002) 35
7. Fellbaum, C., Palmer, M., Dang, H.T., Delfs, L., Wolff, S.: Manual and Automatic Semantic Annotation with WordNet. In: *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources: Applications, Customizations*. Carnegie Mellon University. Pittsburg, PA (2001) 33
8. Habash, N., Dorr, B.J.: Generation-Heavy Machine Translation. In: *Proceedings of the Fifth Conference of the Association for Machine Translation in the Americas, AMTA-2002* (this volume). Tiburon, CA. (2002) 33, 41
9. Han, C.-H., Lavoie, B., Palmer, M., Rambow, O., Kittredge, R., Korelsky, T., Kim, N., Kim, M.: Handling Structural Divergences and Recovering Dropped Arguments in a Korean/English Machine Translation System. In: *Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas, AMTA-2000*. Cuernavaca, Mexico (2000) 34
10. Hermjakob, U., Mooney, R.J.: Learning Parse and Translation Decisions from Examples with Rich Context. In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. (1997) 482-489 33
11. Hwa, R.: Sample selection for statistical grammar induction. In: *Proceedings of the 2000 Joint SIGDAT Conference on EMNLP and VLC*. Hong Kong, China (2000) 45-52 33
12. Hwa, R., Resnik, P., Weinberg, A., Kolak, O.: Evaluating Translational Correspondence Using Annotation Projection. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA (2002) 37, 41

13. Lavoie, B., Kittredge, R., Korelsky, T., Rambow, O.: A Framework for MT and Multilingual NLG Systems Based on Uniform Lexico-Structural Processing. In: Proceedings of the 1st Annual North American Association of Computational Linguistics, ANLP/NAACL-2000. Seattle, WA (2000) 34
14. Lavoie, B., White, M., Korelsky, T.: Inducing Lexico-Structural Transfer Rules from Parsed Bi-texts. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics – DDMT Workshop. Toulouse, France (2001) 34
15. Lin, D.: Government-Binding Theory and Principle-Based Parsing. University of Maryland Technical Report. Submitted to Computational Linguistics. University of Maryland (1995) 33
16. Lin, D.: Dependency-Based Evaluation of MINIPAR. In: Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation. Granada, Spain (1998) 33
17. Melamed, I.D.: Empirical Methods for MT Lexicon Development. In: Proceedings of the Third Conference of the Association for Machine Translation in the Americas, AMTA-98. Langhorne, PA (1998) 38
18. Menezes, A., Richardson, S.D.: A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics – DDMT Workshop. Toulouse, France (2001) 34
19. Meyers, A., Kosaka, M., Grishman, R.: Chart-Based Transfer Rule Application in Machine Translation. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000). Saarbrücken, Germany (2000) 34
20. Och, F.J., Ney, H.: Improved Statistical Alignment Models. In: Proceedings of the 38th Annual Conference of the Association for Computational Linguistics. Hongkong, China (2000) 440-447 33
21. Slobin, D.I.: Two Ways to Travel: Verbs of Motion in English and Spanish. In: Shibatani, M., Thompson, S.A. (eds.): Grammatical Constructions: Their Form and Meaning. Oxford University Press, New York (1996) 195-219 38
22. Watanabe, H., Kurohashi, S., Aramaki, E.: Finding Structural Correspondences from Bilingual Parsed Corpus for Corpus-based Translation. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000). Saarbrücken, Germany (2000) 34
23. Wu, D.: Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. Computational Linguistics. Vol. 23(3) (1997) 377-400 33
24. Xia, F., Palmer, M., Xue, N., Okurowski, M.E., Kovarik, J., Huang, S., Kroch, T., Marcus, M.: Developing Guidelines and Ensuring Consistency for Chinese Text Annotation. In: Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-2000). Athens, Greece (2000) 41
25. Yamada, K., Knight, K.: A Syntax-Based Statistical Translation Model. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics. Toulouse, France (2001) 523-529 31, 33
26. Yarowsky, D., Ngai, G.: Inducing Multilingual POS Taggers and NP Brackets via Robust Projection across Aligned Corpora. In: Proceedings of NAACL-2001. Pittsburgh, PA (2001) 200-207 31

Text Prediction with Fuzzy Alignments

George Foster, Philippe Langlais, and Guy Lapalme

RALI, Université de Montréal
`www-rali.iro.umontreal.ca`

Abstract. Text prediction is a form of interactive machine translation that is well suited to skilled translators. In recent work it has been shown that simple statistical translation models can be applied within a user-modeling framework to improve translator productivity by over 10% in simulated results. For the sake of efficiency in making real-time predictions, these models ignore the alignment relation between source and target texts. In this paper we introduce a new model that captures fuzzy alignments in a very simple way, and show that it gives modest improvements in predictive performance without significantly increasing the time required to generate predictions.

1 Introduction

The idea of using text prediction as a tool for translators was first introduced by Church and Hovy as one of many possible applications for “crummy” machine translation technology [2]. Text prediction can be seen as a form of interactive MT that is well suited to skilled translators. Compared to the traditional form of IMT based on Kay’s original work [7]—in which the user’s role is to help disambiguate the source text—prediction is less obtrusive and more natural, allowing the translator to focus on and directly control the contents of the target text. Predictions can benefit a translator in several ways: by accelerating typing, by suggesting translations, and by serving as an implicit check against errors.

The first implementation of a predictive tool for translators was described in [3], in the form of a simple word-completion system based on statistical models. Various enhancements to this were carried out as part of the TransType project [9], including the addition of a realistic user interface, better models, and the capability of predicting multi-word lexical units. In the final TransType prototype for English to French translation, the translator is presented with a short pop-up menu of predictions after each character typed. These may be incorporated into the text with a special command or rejected by continuing to type normally.

Although TransType is capable of correctly anticipating over 70% of the characters in a freely-typed translation (within the domain of its training corpus), this does not mean that users can translate in 70% less time when using the tool. In fact, in a trial with skilled translators, the users’ rate of text production *declined* by an average of 17% as a result of using TransType [10]. There are two main reasons for this. First, it takes time to read the system’s proposals, so that in cases where they are wrong or too short, the net effect will be to slow

the translator down. Second, translators do not always act “rationally” when confronted with a proposal; that is, they do not always accept correct proposals and they occasionally accept incorrect ones.

In previous work [6], we described a new approach to text prediction intended to address these problems. The main idea is to make proposals that maximize the expected benefit to the user in each context, rather than systematically predicting a fixed amount of text after each character typed. The expected benefit is estimated from two components: a statistical translation model that gives the probability that a candidate prediction will be correct or incorrect, and a user model that determines the benefit to the translator in either case. Simulated results indicate that this approach has the potential to increase translator productivity by over 10%, a considerable improvement over the -17% observed in the TransType trials.

For the sake of efficiency in making real-time predictions, the statistical translation model used in [6] ignores the alignment relation between source and target texts. Although this has negligible effect on very short predictions (for instance completions for the current word), it is noticeable in longer predictions, which occasionally repeat previous segments of target text or contain translations for words that have already been translated. In this paper, we introduce and evaluate a new translation model that adds a notion of fuzzy alignments to the maximum-entropy model of [6].

The structure of the paper is as follows. Section 2 outlines the basic approach to text prediction. The subsequent three sections describe the main elements of this approach: translation model, user model, and search procedure (the latter two are condensed versions of the descriptions given in [6]). The last two sections give results and conclude.

2 The Text Prediction Task

In the basic prediction task, the input to the predictor is a source sentence $\mathbf{s}$ and a prefix $\mathbf{h}$ of its translation (ie, the target text before the current cursor position); the output is a proposed extension $\mathbf{x}$ to $\mathbf{h}$. Figure 1 gives an example. Unlike the TransType prototype, which proposes a set of alternate single-word suggestions, each prediction here consists of only a single proposal, but one that may span an arbitrary number of words.

As described above, the goal of the predictor is to find the prediction $\hat{\mathbf{x}}$ that maximizes the expected benefit to the user:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} B(\mathbf{x}, \mathbf{h}, \mathbf{s}), \quad (1)$$

where $B(\mathbf{x}, \mathbf{h}, \mathbf{s})$ measures typing time saved. This obviously depends on how much of $\mathbf{x}$ is correct, and how long it would take to edit it into the desired text. A major simplifying assumption we make is that the user edits only by erasing wrong characters from the end of a proposal. Given a TransType-style interface where acceptance places the cursor at the end of a proposal, this is

s: Let us return to serious matters.

t: $\overbrace{\text{On va r}}^{\mathbf{h}} \overbrace{\text{evenir aux choses sérieuses.}}^{\mathbf{x}^*}$

x: *evenir à*

Fig. 1. Example of a prediction for English to French translation. $\mathbf{s}$ is the source sentence, $\mathbf{h}$ is the part of its translation that has already been typed, $\mathbf{x}^*$ is what the translator wants to type, and $\mathbf{x}$ is the prediction.

the most common editing method, and it gives a conservative estimate of the cost attainable by other methods. With this assumption, the key determinant of edit cost is the length of the correct prefix of $\mathbf{x}$, so the expected benefit can be written as:

$$B(\mathbf{x}, \mathbf{h}, \mathbf{s}) = \sum_{k=0}^l p(k|\mathbf{x}, \mathbf{h}, \mathbf{s}) B(\mathbf{x}, \mathbf{h}, \mathbf{s}, k), \quad (2)$$

where $p(k|\mathbf{x}, \mathbf{h}, \mathbf{s})$ is the probability that exactly k characters from the beginning of $\mathbf{x}$ will be correct, l is the length of $\mathbf{x}$, and $B(\mathbf{x}, \mathbf{h}, \mathbf{s}, k)$ is the benefit to the user given that the first k characters of $\mathbf{x}$ are correct.

Equations (1) and (2) define three main problems: estimating the prefix probabilities $p(k|\mathbf{x}, \mathbf{h}, \mathbf{s})$, estimating the user benefit function $B(\mathbf{x}, \mathbf{h}, \mathbf{s}, k)$, and searching for $\hat{\mathbf{x}}$.

3 Translation Models

The correct-prefix probabilities $p(k|\mathbf{x}, \mathbf{h}, \mathbf{s})$ are derived from a statistical translation model that gives the probability $p(w|\mathbf{h}, \mathbf{s})$ that a word w will follow a previous sequence of words $\mathbf{h}$ in the translation of $\mathbf{s}$. (Note that this model does not distinguish between words in $\mathbf{h}$ that have been sanctioned by the translator and those that are hypothesized as part of the search procedure to find the best prediction.) As described in [6], the derivation involves first converting prefix probabilities to explicit character string probabilities of the form $p(x_1^k|\mathbf{h}, \mathbf{s})$, then calculating these by summing over all compatible token sequences.

3.1 MEMD2B

The model for $p(w|\mathbf{h}, \mathbf{s})$ used in [6] is a maximum entropy/minimum divergence (MEMD) model of the form:

$$p(w|\mathbf{h}, \mathbf{s}) = \frac{q(w|\mathbf{h}) \exp(\sum_{s \in \mathbf{s}} \alpha_{sw} + \alpha_{D(s, w, i, j)})}{Z(\mathbf{h}, \mathbf{s})},$$

where $q(w|\mathbf{h})$ is a trigram language model; α_{sw} is a weight that captures the strength of the association between the current target word w and a word s in the

source sentence $\mathbf{s}$; $\alpha_{D(s,w,i,\hat{j})}$ is a weight that depends on the distance between the position i of w , and the position $\hat{j}$ of the closest occurrence of s in $\mathbf{s}$; and $Z(\mathbf{h}, \mathbf{s})$ is a normalizing factor (the sum over all w of the numerator). The parameters of the model are the families of weights α_{sw} (one for each selected bilingual word pair), and $\alpha_{D(s,w,i,\hat{j})}$ (one for each position/word-pair class), which are set so as to maximize the likelihood of a training corpus. More details are given in [5], where this model is labelled MEMD2B.

MEMD2B is analogous to a linear combination of a trigram and the IBM model 2 [1], as used in the TransType prototype:

$$p(w|\mathbf{h}, \mathbf{s}) = \lambda q(w|\mathbf{h}) + (1 - \lambda) \sum_{j=0}^J p(w|s_j)p(j|i, J)$$

where $\lambda \in [0, 1]$ is a combining weight, J is the number of tokens in $\mathbf{s}$, $p(w|s_j)$ is a translation probability that plays a role similar to the MEMD word-pair weight α_{sw} , and $p(j|i, J)$ is a position probability that plays a role similar to the MEMD position weight $\alpha_{D(s,w,i,\hat{j})}$. The most significant differences between the MEMD and linear models are that MEMD model combines the contributions from its language and translation components by multiplying instead of adding them; and that the MEMD translation parameters are learned in the presence of the trigram language model. These characteristics make MEMD2B significantly more powerful than its linear counterpart, even when it is defined using an order of magnitude fewer parameters. It yields about 50% lower test-corpus perplexity [5], and about 69% higher keystroke savings on the text prediction task (see table 2 in [6]—this is using the “best” estimates for both models, when predictions are limited to 5 words or fewer).

Both the MEMD and linear models have an obvious weakness, in that their translation components depend only on the length of $\mathbf{h}$ (ie, the position of w), and not on the actual words it contains. From the standpoint of predictive efficiency, this is a good thing, since it means that the models support $O(mJV^3)$ Viterbi-style searches for the most likely sequence of m words that follows $\mathbf{h}$, where V is the size of the target-language vocabulary. However, this speed comes at the cost of accuracy, because only the trigram and the relatively weak contribution from the position parameters prevent the models from assigning high probabilities to words that are repeat or alternate translations of source words that already have translations in $\mathbf{h}$. To ensure that this does not happen, a model must capture the alignment relation between $\mathbf{h}$ and $\mathbf{s}$ in some way.

3.2 Noisy Channel

In statistical MT, the standard way to capture the alignment relation is a noisy-channel approach, where the natural distribution $p(\mathbf{t}|\mathbf{s})$ for a target text $\mathbf{t}$ given a source text $\mathbf{s}$ is modeled via Bayes law as proportional to $p(\mathbf{s}|\mathbf{t})p(\mathbf{t})$ for a fixed source text. This combines language and translation components in an optimum way, and allows even the simplest IBM translation models 1 and 2 to capture the

alignment relation. The drawback is that the decoding problem is NP-complete for noisy-channel models [8], and even the fastest dynamic-programming heuristics used in statistical MT [11,12], are polynomial in J —for instance $O(mJ^4V^3)$ in [12].

For the prediction application, a noisy channel decomposition of the distribution $p(w|\mathbf{h}, \mathbf{s})$ is:

$$p(w|\mathbf{h}, \mathbf{s}) = p(w|\mathbf{h})p(\mathbf{s}|\mathbf{h}, w)/p(\mathbf{s}|\mathbf{h}), \quad (3)$$

where, unlike in SMT, the denominator $p(\mathbf{s}|\mathbf{h})$ must be retained in order to give true probabilities for the user benefit calculation. To ensure that the resulting distribution is normalized over all w , $p(\mathbf{s}|\mathbf{h})$ can be calculated by summing the numerator over all words in the vocabulary:

$$p(\mathbf{s}|\mathbf{h}) = \sum_w p(w|\mathbf{h})p(\mathbf{s}|\mathbf{h}, w).$$

To see how the noisy-channel approach enforces alignment constraints, consider an IBM1 model for $p(\mathbf{s}|\mathbf{h}, w)$:

$$p(\mathbf{s}|\mathbf{h}, w) = \prod_{j=1}^J \left[\sum_{i=0}^I p(s_j|h_i) + p(s_j|w) \right] / (I+2)$$

where I is the length of $\mathbf{h}$, and h_i is the i th word in $\mathbf{h}$. Dividing by the constant factor $\prod_{j=1}^J \sum_{i=0}^I p(s_j|h_i)/(I+2)$ gives:

$$p(\mathbf{s}|\mathbf{h}, w) \propto \prod_{j=1}^J \left[1 + \frac{p(s_j|w)}{\sum_{i=0}^I p(s_j|h_i)} \right].$$

So the score assigned to w is a product of source word scores, each of which can only be significantly greater than 1 if the probability that the corresponding source word s translates to w is significantly larger than the sum of the probabilities of all previous translations for s in $\mathbf{h}$. As a consequence, the probability assigned to the first good translation of any source word will be much higher than that assigned to subsequent translations of the same word.

Unfortunately, apart from the expensive search properties noted above, the noisy channel model described here does not seem optimal for word prediction with a user model. The reason is that the distribution over w it gives is very flat, yielding probabilities for the best next words that are lower than their true probabilities (which are crucial for this application). This is reflected in the test corpus perplexity of the model, which is only slightly lower than that of the trigram language model on its own (even when IBM2 is used as the translation component). Tuning the model with various parameters such as an exponential weight on the translation component, or weights on the initial source-word scores $p(s_j|h_0)$, does not substantially improve this picture.

3.3 Fuzzy Alignments

Another approach to enforcing alignment constraints is to build them into the existing MEMD2B model. One way to do this would be to add features to capture translation relations between $\mathbf{h}$ and $\mathbf{s}$. We have explored a simpler alternate approach based on modulating the weights of existing features that are active on the words in $\mathbf{h}$ —essentially “ticking-off” source words that appear to have valid translations in $\mathbf{h}$. Our starting point is the observation that the word pairs captured by MEMD2B can be divided into two categories according to the magnitude of their weights, as shown in table 1: pairs with large weights tend to be true translations, while those with small weights tend to be looser semantic and grammatical associations.¹

We experimented with a number of simple ways of exploiting this distinction, and found that the algorithm shown in figure 2 worked best. This uses a threshold f_1 to classify word-pairs as either translations or associations. Associations that occur between word pairs in $\mathbf{h}$ and $\mathbf{s}$ are modulated by a parameter f_2 , while translation weights are set to the parameter f_3 . For each pair deemed a translation, all weights involving the source word are modulated by f_4 . Values for all four f_i parameters were optimized on a cross-validation corpus, using perplexity as a criterion.

This algorithm is applied sequentially to the target words in $\mathbf{h}$, and its effects are incremental. Although the results will in general depend on the order in which target words are processed, no attempt is made to find the optimal order—processing is always left-to-right. We expect this dependence to be weak in any case, as the intention is to capture alignments in a fuzzy way, accounting for all possibilities at once and avoiding an expensive search for the optimal alignment.

Table 1. Five smallest positive (top box) and five largest (bottom box) word-pair weights for the MEMD2B model.

this	ne	0.000164881
home	circonscription	0.000180333
very	le	0.000299113
,	aussi	0.000300543
at	dans	0.000360243
c-283	c-283	11.9133
mid-november	mi-novembre	11.9148
732	732	11.9304
darryl	darryl	11.9383
c-304	c-304	11.9559

¹ Many of these appear to be spurious, but they capture statistically valid relationships within the domain—if they are eliminated from the model, its performance on new text within the domain drops.

```

for each source word  $s \in \mathbf{s}$ :
  if  $\alpha_{sw} < f_1$ :
    set  $\alpha_{sw} \leftarrow \alpha_{sw} f_2$ 
  else:
    set  $\alpha_{sw} \leftarrow f_3$ 
  for all target words  $w' \neq w$ 
    set  $\alpha_{sw'} \leftarrow \alpha_{sw'} f_4$ 

```

Fig. 2. Algorithm for modulating MEMB2B word-pair weights to account for the presence of some target word w in $\mathbf{h}$.

4 User Model

The purpose of the user model is to determine the expected benefit $B(\mathbf{x}, \mathbf{h}, \mathbf{s}, k)$ to the translator of a prediction $\mathbf{x}$ whose first k characters match the text that the translator wishes to type. This will depend heavily on whether the translator decides to accept or reject the prediction, so the first step in our model is the following expansion:

$$B(\mathbf{x}, \mathbf{h}, \mathbf{s}, k) = \sum_{a \in \{0,1\}} p(a|\mathbf{x}, \mathbf{h}, \mathbf{s}, k) B(\mathbf{x}, \mathbf{h}, \mathbf{s}, k, a),$$

where $p(a|\mathbf{x}, \mathbf{h}, \mathbf{s}, k)$ is the probability that the translator accepts or rejects $\mathbf{x}$, $B(\mathbf{x}, \mathbf{h}, \mathbf{s}, k, a)$ is the benefit they derive from doing so, and a is a random variable that takes on the values 1 for acceptance and 0 for rejection. The first two quantities are the main elements in the user model, and are described in following sections. The parameters of both were estimated from data collected during the TransType trial described in [10], which involved nine accomplished translators using a prototype prediction tool for approximately half an hour each. In all cases, estimates were made by pooling the data for all nine translators.

4.1 Acceptance Probability

The model for $p(a|\mathbf{x}, \mathbf{h}, \mathbf{s}, k)$ is based on the assumption that the probability of accepting $\mathbf{x}$ depends on roughly what the user stands to gain from it, defined according to the editing scenario given in section 2 as the amount by which the length of the correct prefix of $\mathbf{x}$ exceeds the length of the incorrect suffix:

$$p(a|\mathbf{x}, \mathbf{h}, \mathbf{s}, k) \approx p(a|2k - l),$$

where $k - (l - k) = 2k - l$ is called the *gain*. For instance, the gain for the prediction in figure 1 would be $2 \times 7 - 8 = 6$. It is straightforward to make empirical estimates of acceptance probabilities for each gain value; the model is simply a smoothed curve fit to these points.

4.2 Benefit

The benefit $B(\mathbf{x}, \mathbf{h}, \mathbf{s}, k, a)$ is defined as the typing time the translator saves by accepting or rejecting a prediction $\mathbf{x}$ whose first k characters are correct. To estimate this, we assume that the translator first reads $\mathbf{x}$, then, if he or she decides to accept, uses a special command to place the cursor at the end of $\mathbf{x}$ and erases its last $l - k$ characters. Assuming independence from $\mathbf{h}, \mathbf{s}$ as before, our model is:

$$B(\mathbf{x}, k, a) = \begin{cases} -R_1(\mathbf{x}) + T(\mathbf{x}, k) - E(\mathbf{x}, k), & a = 1 \\ -R_0(\mathbf{x}), & a = 0 \end{cases}$$

where $R_a(\mathbf{x})$ is the cost of reading $\mathbf{x}$ when it ultimately gets accepted ($a = 1$) or rejected ($a = 0$), $T(\mathbf{x}, k)$ is the cost of manually typing x_1^k , and $E(\mathbf{x}, k)$ is the edit cost of accepting $\mathbf{x}$ and erasing to the end of its first k characters. All of these elements are converted to units of keystrokes saved: $T(\mathbf{x}, k)$ and $E(\mathbf{x}, k)$ are estimated as k and $l - k + 1$ respectively; and read costs are converted from average elapsed times from proposal display to the next user action.

5 Search

Searching directly through all character strings $\mathbf{x}$ in order to find $\hat{\mathbf{x}}$ according to equation (1) would be very expensive. The fact that $B(\mathbf{x}, \mathbf{h}, \mathbf{s})$ is non-monotonic in the length of $\mathbf{x}$ makes it difficult to organize efficient dynamic-programming search techniques or use heuristics to prune partial hypotheses. Because of this, we adopted a fairly radical search strategy that involves first finding the most likely sequence of words of each length, then calculating the benefit of each of these sequences to determine the best proposal. The algorithm is:

1. For each length $m = 1 \dots M$, find the best word sequence:

$$\hat{\mathbf{w}}_m = \operatorname{argmax}_{w_1^m} p(w_1^m | \mathbf{h}, \mathbf{s}),$$

2. Convert each $\hat{\mathbf{w}}_m$ to a corresponding character string $\hat{\mathbf{x}}_m$.
3. Output $\hat{\mathbf{x}} = \operatorname{argmax}_m B(\hat{\mathbf{x}}_m, \mathbf{h}, \mathbf{s})$, or the empty string if all $B(\hat{\mathbf{x}}_m, \mathbf{h}, \mathbf{s})$ are non-positive.

Step 1 is carried out using a Viterbi beam search with the translation model $p(w|\mathbf{h}, \mathbf{s})$. To speed this up, the search is limited to an *active vocabulary* of target words likely to appear in translations of $\mathbf{s}$, defined as the set of all words connected by some word-pair feature in our translation model to some word in $\mathbf{s}$. Step 2 is a trivial deterministic procedure that mainly involves deciding whether or not to introduce blanks between adjacent words (eg yes in the case of *la + vie*, no in the case of *l' + an*). Step 3 involves a straightforward evaluation of m strings according to equation (2).

Table 2 shows empirical search timings for various values of M , for both the baseline MEMD2B model and the alignment version. Although the average times for the alignment model are higher, they are still well below values that would cause delays perceptible to a user.

Table 2. Approximate times in seconds to generate predictions of maximum word sequence length M , on a 1.2GHz processor.

M	MEMD2B		MEMD2B-align	
	average time	maximum time	average time	maximum time
1	0.0012	0.01	0.0014	0.02
2	0.0038	0.23	0.0043	0.25
3	0.0097	0.51	0.0109	0.65
4	0.0184	0.55	0.0209	0.72
5	0.0285	0.57	0.0323	0.73

Table 3. Prediction results. Numbers give estimated percent reductions in keystrokes. Columns give the maximum permitted number of words M in predictions. Rows correspond to different predictor configurations: *fixed* ignores the user model and systematically makes M -word predictions; *standard* optimizes according to the user model, with model probabilities modified by the length-specific correction factors described in [6] (tuned separately for each model); and *best* gives an upper bound obtained by choosing m in step 3 of the search algorithm so as to maximize $B(\hat{\mathbf{x}}_m, \mathbf{h}, \mathbf{s}) = B(\hat{\mathbf{x}}_m, \mathbf{h}, \mathbf{s}, k_m)$, where k_m is the true value of k for $\hat{\mathbf{x}}_m$, from the test corpus.

config	MEMD2B					MEMD2B-align				
	M					M				
	1	2	3	4	5	1	2	3	4	5
fixed	-8.5	-0.4	-3.6	-11.6	-20.8					
standard	5.8	10.7	12.0	12.5	12.6	5.8	10.9	12.7	13.2	13.4
best	7.9	17.9	24.5	27.7	29.2					

6 Evaluation

To test the effect of adding alignment parameters to MEMD2B, we evaluated the English to French prediction performance of both the baseline model and the alignment version using the simulation technique described in [6]. The test corpus consisted of 5,020 Hansard sentence pairs and approximately 100k words in each language; details of the training corpus are given in [4]. The results are shown in table 3. The difference between the two models is negligible for predictions of less than three words, but increasingly significant for longer predictions, reaching a maximum relative improvement for the alignment model of about 6% with a prediction length limit of 5. This is in line with our intuition that the effect of including alignments should be more pronounced for longer predictions.

7 Conclusion

We have described a new maximum-entropy translation model for text prediction that improves on a previous model by incorporating a fuzzy notion of the alignment relation between the source text $\mathbf{s}$ and some initial part $\mathbf{h}$ of its translation. The improved model works at essentially the same speed as the previous one, and gives an increase of about 6% in estimated translator effort saved when

predictions are limited to at most five words. This is a modest improvement, but on the other hand it is achieved by adding only four parameters to the baseline maximum-entropy model. We feel that it demonstrates the potential of fuzzy alignments for this application, and we plan to investigate more sophisticated approaches in the future, possibly involving the addition of dedicated alignment features to the model.

References

1. Brown, P.F., Pietra, S.A.D., Pietra, V.D.J., Mercer, R.L.: The mathematics of Machine Translation: Parameter estimation. *Computational Linguistics* **19** (1993) 263–312 47
2. Church, K.W., Hovy, E.H.: Good applications for crummy machine translation. *Machine Translation* **8** (1993) 239–258 44
3. Foster, G., Isabelle, P., Plamondon, P.: Target-text Mediated Interactive Machine Translation. *Machine Translation* **12** (1997) 175–194 44
4. Foster, G.: A Maximum Entropy / Minimum Divergence translation model. In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, Hong Kong (2000) 52
5. Foster, G.: Incorporating position information into a Maximum Entropy / Minimum Divergence translation model. In: *Proceedings of the 4th Computational Natural Language Learning Workshop (CoNLL)*, Lisbon, Portugal, ACL SigNLL (2000) 47, 47
6. Foster, G., Langlais, P., Lapalme, G.: User-friendly text prediction for translators. In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, PA (2002) 45, 45, 45, 45, 46, 46, 47, 52, 52
7. Kay, M.: The MIND system. In Rustin, R., ed.: *Natural Language Processing*. Algorithmics Press, New York (1973) 155–188 44
8. Knight, K.: Decoding complexity in word-replacement translation models. *Computational Linguistics, Squibs and Discussion* **25** (1999) 48
9. Langlais, P., Foster, G., Lapalme, G.: Unit completion for a computer-aided translation typing system. *Machine Translation* **15** (2000) 267–294 44
10. Langlais, P., Lapalme, G., Loranger, M.: TransType: From an idea to a system. *Machine Translation* (2002) To Appear. 44, 50
11. Niessen, S., Vogel, S., Ney, H., Tillmann, C.: A DP based search algorithm for statistical machine translation. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th International Conference on Computational Linguistics (COLING)* 1998, Montréal, Canada (1998) 960–967 48
12. Tillmann, C., Ney, H.: Word re-ordering and DP-based search in statistical machine translation. In: *Proceedings of the International Conference on Computational Linguistics (COLING)* 2000, Saarbrücken, Luxembourg, Nancy (2000) 48, 48

Efficient Integration of Maximum Entropy Lexicon Models within the Training of Statistical Alignment Models[□]

Ismael García Varea¹, Franz J. Och², Hermann Ney², and Francisco
Casacuberta³

¹ Dpto. de Inf., Univ. de Castilla-La Mancha, 02071 Albacete, Spain
`ivarea@info-ab.uclm.es`

² Lehrstuhl für Inf. VI, RWTH Aachen, Ahornstr., 55 D-52056 Aachen, Germany

³ Inst. Tecnológico de Inf., Univ. Politécnica de Valencia, 46071 Valencia, Spain

Abstract. Maximum entropy (ME) models have been successfully applied to many natural language problems. In this paper, we show how to integrate ME models efficiently within a maximum likelihood training scheme of statistical machine translation models. Specifically, we define a set of context-dependent ME lexicon models and we present how to perform an efficient training of these ME models within the conventional expectation-maximization (EM) training of statistical translation models. Experimental results are also given in order to demonstrate how these ME models improve the results obtained with the traditional translation models. The results are presented by means of alignment quality comparing the resulting alignments with manually annotated reference alignments.

1 Introduction

The ME approach has been applied in natural language processing and machine translation to a variety of tasks. In [1] this approach is applied to the so-called IBM Candide system to build context-dependent models, to compute automatic sentence splitting and to improve word reordering in translation. Similar techniques are used in [10] for so-called direct translation models instead of those proposed in [2]. In [6] ME models are used to reduce translation test perplexities and translation errors by means of a rescoring algorithm, which is applied to n-best translation hypotheses. In [5] two methods for incorporating information about the relative position of bilingual word pairs into a ME translation model are described. Other authors have used this approach to language modeling [11].

In this paper, we show how to integrate ME models efficiently within a maximum likelihood training scheme of statistical machine translation models. Specifically, we define a set of context-dependent ME lexicon models and we describe how to perform an efficient training of these ME models within the conventional

* This work has been partially supported by Spanish CICYT under grant TIC2000-1599-C02-01

EM training of statistical alignment models [2]. In each iteration of the training process, the set of ME models is automatically generated by using the set of possible word-alignments between each pair of sentences. The ME models are trained with the Generalized Iterative Scaling (GIS) algorithm [3], then used in the next iteration of the EM training process in order to recompute a new set of parameters of the translation models.

Experimental results are given for the French–English Canadian Parliament Hansards corpus and the Verbmobil task. The evaluation is performed by comparing the Viterbi alignments obtained after the training of the conventional and the integrated approaches with manually annotated reference alignments.

2 Statistical Machine Translation

The goal of the translation process in statistical machine translation can be formulated as follows: a source language string $\mathbf{f} = f_1^J = f_1 \dots f_J$ is to be translated into a target language string $\mathbf{e} = e_1^I = e_1 \dots e_I$. Every target string is regarded as a possible translation for the source language string with maximum a-posteriori probability $Pr(\mathbf{e}|\mathbf{f})$. According to Bayes' decision rule, we have to choose the target string that maximizes the product of both the target language model $Pr(\mathbf{e})$ and the string translation model $Pr(\mathbf{f}|\mathbf{e})$. Alignment models to structure the translation model are introduced in [2]. These alignment models are similar to the concept of Hidden Markov models (HMM) in speech recognition. The alignment mapping is $j \rightarrow i = a_j$ from source position j to target position $i = a_j$. In statistical alignment models, $Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$, the alignment $\mathbf{a}$ is introduced as a hidden variable.

The translation probability $Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ can be rewritten as follows:

$$\begin{aligned} Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) &= \prod_{j=1}^J Pr(f_j, a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \\ &= \prod_{j=1}^J \left(Pr(a_j | f_1^{j-1}, a_1^{j-1}, e_1^I) \cdot Pr(f_j | f_1^{j-1}, a_1^j, e_1^I) \right) . \end{aligned} \quad (1)$$

3 Conventional EM Training (Review)

In this section, we describe the training of the model parameters. Every model has a specific set of free parameters. For example, the parameters θ for Model 4 [2] consist of lexicon, alignment and fertility parameters:

$$\theta = \{ \{p(f|e)\}, \{p_{=1}(\Delta j)\}, \{p_{>1}(\Delta j)\}, \{p(\phi|e)\}, p_1 \} . \quad (2)$$

To train the model parameters θ , we pursue a maximum likelihood approach using a parallel training corpus consisting of S sentence pairs $\{(\mathbf{f}_s, \mathbf{e}_s) : s = 1, \dots, S\}$:

$$\hat{\theta} = \arg \max_{\theta} \prod_{s=1}^S \sum_{\mathbf{a}} p_{\theta}(\mathbf{f}_s, \mathbf{a}|\mathbf{e}_s) . \quad (3)$$

We do this by applying the EM algorithm. The different models are trained in succession on the same data, where the final parameter values of a simpler model serve as the starting point for a more complex model.

In the E-step, the lexicon parameter counts for one sentence pair $(\mathbf{f}, \mathbf{e})$ are calculated:

$$c(f|e; \mathbf{f}, \mathbf{e}) = N(\mathbf{f}, \mathbf{e}) \cdot \sum_{\mathbf{a}} Pr(\mathbf{a}|\mathbf{f}, \mathbf{e}) \sum_j \delta(f, f_j) \delta(e, e_{a_j}) . \quad (4)$$

Here, $N(\mathbf{f}, \mathbf{e})$ is the training corpus count of the sentence pair $(\mathbf{f}, \mathbf{e})$. In the M-step, we want to compute the lexicon parameters $p(f|e)$ that maximize the likelihood on the training corpus. This results in the following re-estimation [2]:

$$p(f|e) = \frac{\sum_s c(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})}{\sum_{s,f} c(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})} . \quad (5)$$

Similarly, the alignment and fertility probabilities can be estimated for all other alignment models [2]. When bootstrapping from a simpler model to a more complex model, the simpler model is used to weight the alignments and the counts are accumulated for the parameters of the more complex model.

4 Maximum Entropy Modeling

4.1 Motivation

Typically, the probability $Pr(f_j | f_1^{j-1}, a_1^j, e_1^I)$ in Equation (1) is approximated by a lexicon model $p(f_j | e_{a_j})$ by dropping the dependencies on f_1^{j-1} , a_1^{j-1} , $e_1^{a_j \otimes 1}$, and $e_{a_{j+1}}^I$. Obviously, this simplification is not true for many natural language phenomena. The straightforward approach to include more dependencies in the lexicon model would be to add additional dependencies (e.g. $p(f_j | e_{a_j}, e_{a_j \otimes 1})$). This approach would yield a significant data sparseness problem. For this reason, we define a set of context-dependent ME lexicon models, which is directly integrated into a conventional EM training of the statistical alignment models.

In this case, the role of ME is to build a stochastic model that efficiently takes a larger context into account. In the remainder of the paper, we shall use $p_e(f|x)$ to denote the probability that the ME model (which is associated to e) assigns to f in the context x .

4.2 Maximum Entropy Principle

In the ME approach, we describe all properties that we deem to be useful by so-called feature functions $\phi_{e,k}(x, f)$, $k = 1, \dots, K_e$. For example, let us suppose that the k -th feature for word e tries to model the existence or absence of a specific word e'_k in the context of an English word e , which can be translated by f'_k . We can express this dependence using the following feature function:

$$\phi_{e,k}(x, f) = \begin{cases} 1 & \text{if } f = f'_k \text{ and } e'_k \in x \\ 0 & \text{otherwise} \end{cases} . \quad (6)$$

5 Integrated EM–ME Training

5.1 Training Integration

Using a ME lexicon model for a target word e , we have to train the model parameters $\Lambda_e \equiv \{\lambda_{e,k} : k = 1, \dots, K_e\}$ instead of the parameters $\{p(f|e)\}$. We pursue the following approach. In the E-step, we perform a refined count collection for the lexicon parameters:

$$c(f|e, x; \mathbf{f}, \mathbf{e}) = N(\mathbf{f}, \mathbf{e}) \cdot \sum_{\mathbf{a}} Pr(\mathbf{a}|\mathbf{f}, \mathbf{e}) \sum_j \delta(f, f_j) \delta(e, e_{a_j}) \delta(x, x_{j,a_j}) . \quad (8)$$

Here, x_{j,a_j} should denote the ME context that surrounds f_j and e_{a_j} . In the M-step, we want to compute the lexicon parameters that maximize the likelihood:

$$\hat{\Lambda}_e = \arg \max_{\Lambda_e} \prod_{f,x} c(f|e, x; \mathbf{f}, \mathbf{e}) \cdot \log p_e(f|x) . \quad (9)$$

Hence, the refined lexicon counts $c(f|e, x; \mathbf{f}, \mathbf{e})$ are the weights of the set of training samples (f, e, x) , which are used to train the ME models.

The re-estimation of the alignment and fertility probabilities does not change if we use a ME lexicon model.

Thus, we obtain the following steps of each iteration for the EM algorithm:

1. E-step:
 - (a) Collect counts for alignment and fertility parameters.
 - (b) Collect refined lexicon counts and ME training events generation.
2. M-step:
 - (a) Re-estimate alignment and fertility parameters.
 - (b) Perform GIS training for lexicon parameters.

With respect to a conventional EM loop, the steps 1b and 2b involve an overhead on space and computation time. In the next subsection we outline how to address these overheads to make the integrated training as efficient as possible.

5.2 Efficient Training

As an introduction to the integration of the ME training within the EM algorithm let us suppose that we are in the k -th iteration of the EM process. In the E-step of this iteration, for every sentence pair in the training corpus (f_1^J, e_1^I) and every possible alignment between them, we need to use the ME lexicon probability of every word pair (f_j, e_i) computed in the $(k - 1)$ -th iteration. We will then need, a priori, to recompute $p_{e_i}(f_j|x)$ for every computation of $Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ (and also J times for the J words). To perform this computation efficiently, we precompute all possible $p_{e_i}(f_j|x)$ in a translation matrix $(I \times J)$ of ME lexicon probabilities. Thus, each time we need to compute this probability we only need to access the corresponding matrix element. Also in the E-step, the specific ME training events (f_j, e_i, x) generation of current iteration is performed.

After the E-step is carried out for each sentence pair in the corpus, we have all possible ME events (f, e, x) for each word e . Then, with these events, in the M-step we perform a GIS training for every e word we considered (a priori) relevant to our problem and obtain the set of Λ_e parameters that define our specific ME model.

Specially, the following factors contribute to the computational overhead introduced by the ME lexicon models:

1. The computation time of the translation matrix. This involves one order of magnitude of increase.
2. The computation of the GIS training for each word e to be modeled by ME; in the worst case, when all words $e \in \mathcal{V}_e$ (vocabulary of e) are used. In the experiments, the computation time of the GIS algorithm ranges from 5 to 10 seconds on average. This could yield an increase of two orders of magnitude depending on the number of ME models to be considered.

Hence, the additional time consumption depends directly on the number of words e to be modeled by ME. As was described at the end of Section 4.3 we develop a ME model for those words that appear in the training corpus more than a fixed number of times. In our experiments, this word selection yields only about 5-10% of the vocabulary.

With respect to the space overhead, we will need to store every possible ME training event (f, e, x) , that is, every possible combination of $e \in \mathcal{V}_e$, $f \in \mathcal{V}_f$ and x . Obviously this requires a huge quantity of memory, as the word selection described above plays also an important role in the efficiency for space overhead.

The number of training events is an important factor for the computational overhead of the GIS algorithm. Therefore, a pruning of these training events is also applied. As described in the previous subsection, the refined lexicon count (*fractional counts* [2]) are the weights of the ME training events. We prune those training events with fractional counts smaller than 1. Very rare events are thereby discarded and the ME training is faster and better parameter estimation is performed. In addition, the space overhead is also reduced.

In the following, we suggest a simplified approach which reduces the overhead required by this approach. First, we perform a normal training of the EM algorithm. Then, after the final iteration, we perform the ME training of the ME lexicon parameters but use only the Viterbi alignment of each sentence pair instead of the set of all possible alignments. Finally, a new EM training is performed where the lexicon parameters are fixed to the ME lexicon models obtained previously. In this case the more informative contextual information is also used but in a decoupled way. It is important to stress that in this approximation only one ME training is needed. Interestingly, the alignment quality obtained with this simplification and the fully integrated approach are practically the same.

In Table 2, the time consumption in seconds per iteration of the EM algorithm of the different approaches are shown.

Table 2. Time consumption in seconds of different approaches per EM iteration (on average for the five IBM models). # of e means the number of target words to be modeled by ME after the counting-based word selection.

Task	Size of train.	# of e	Normal train	ME-train	Simplified ME-train
Verbmobil	0.5K	29	1	29	1.5
	8K	84	18	235	68
	35K	209	60	2290	675
Hansards	0.5K	15	2.5	29	3
	8K	80	35	1180	100
	128K	1214	655	16890	6870

6 Experimental Results

We present results on the Verbmobil task and the Hansards task. The Verbmobil task is a speech translation task in the domain of appointment scheduling, travel planning, and hotel reservation. The task is difficult because it consists of spontaneous speech and the syntactic structures of the sentences are less restricted and highly variable. The French–English Hansards task consists of the debates in the Canadian Parliament. This task has a very large vocabulary of more than 100,000 French words.

The corpus statistics are shown in Table 3. The number of running words and the vocabularies are based on full-form words including the punctuation marks. We produced smaller training corpora by randomly choosing 500, 8000 and 34000 sentences from the Verbmobil task and 500, 8000 and 128000 sentences from the Hansards task.

To train the context-dependent statistical alignment models, we extended the publicly available toolkit GIZA++ [8]. The training of the ME models was carried out using the YASMET toolkit [8].

6.1 Evaluation Methodology

We use the same annotation scheme for single-word based alignments and a corresponding evaluation criterion as described in [9]. The annotation scheme explicitly allows for ambiguous alignments. The people performing the annotation are asked to specify two different kinds of alignments: an $S(ure)$ alignment, which is used for alignments that are unambiguous and a $P(ossible)$ alignment, which is used for ambiguous alignments. The P label is used particularly to align words within idiomatic expressions, free translations, and missing function words ($S \subseteq P$).

The reference alignment thus obtained may contain many-to-one and one-to-many relationships. Figure 1 shows two examples (of the Hansards task) of manually aligned sentences with S and P labels.

The quality of an alignment $A = \{(j, a_j) | a_j > 0\}$ is then computed by appropriately redefined precision and recall measures and the alignment error

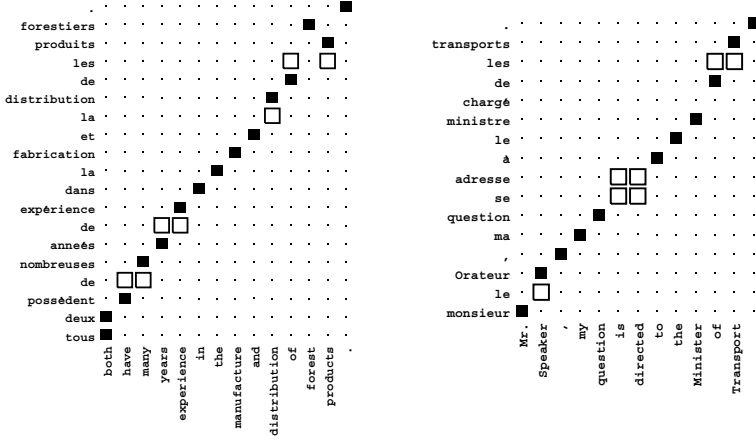


Fig. 1. Two examples of manual alignments with *S(ure)* (■) and *P(ossible)* (□) connections.

rate, which is derived from the well known F-measure:

$$recall = \frac{|A \cap S|}{|S|}, \quad precision = \frac{|A \cap P|}{|A|}, \quad AER(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

Thus, a recall error can only occur if a *S(ure)* alignment is not found. A precision error can only occur if the alignment found is not even *P(ossible)*.

The set of sentence pairs, for which the manual alignment is produced, is randomly selected from the training corpus. It should be emphasized that all the training is done in a completely unsupervised way, i.e. no manual alignments are used. From this point of view, there is no need to have a separate test corpus.

Table 3. Corpus characteristics.

	Verbmobil		Hansards	
	German	English	French	English
Train Sentences	34446		1470K	
Words	329625	343076	24.33M	22.16M
Vocabulary	5936	3505	100269	78332

6.2 Alignment Quality Results

According to the experiments we have carried out so far the differences on alignment quality of the ME integration training with respect to the simplification proposed at the end of Section 5.2 are small. Taking into account the high time consumption of the ME integration, the results presented below are computed by using the simplified approach.

Table 4 shows the alignment quality for different training sample sizes of the Hansards and Verbmobil tasks. This table shows the baseline AER for different

Table 4. AER [%] on Hansards (left) and Verbmobil (right) tasks.

		Size of training corpus							
Train. scheme	Model		0.5K	8K	128K		0.5K	8K	34K
1^5	1	Hansards	48.0	35.1	29.2	Verbmobil	27.7	19.2	17.6
	1+ME		47.7	32.7	22.5		24.6	16.6	13.7
$1^5 2^5$	2		46.0	29.2	21.9		26.8	15.7	13.5
	2+ME		44.7	28.0	19.0		25.3	14.1	10.8
$1^5 2^5 3^3$	3		43.2	27.3	20.8		25.6	13.7	10.8
	3+ME		42.5	26.4	17.2		24.1	11.6	8.8
$1^5 2^5 3^3 4^3$	4		41.8	24.9	17.4		23.6	10.0	7.7
	4+ME		41.3	24.3	14.1		22.8	9.3	7.0
$1^5 2^5 3^3 4^3 5^3$	5		41.5	24.8	16.2		22.6	9.9	7.2
	5+ME		41.2	24.3	14.3		22.3	9.6	6.8

training schemes and the corresponding values when the integration of the ME is done. The training scheme is defined in accordance with the number of iterations performed for each model (4^3 means 3 iterations of Model 4).

The recall and precision results for the Hansards task with and without ME training are shown in Figure 2.

We observe that the alignment error rate improves when using the context-dependent lexicon models. For the Verbmobil task, the improvements were smaller than for the Hansards task, which might be due to the fact that the baseline alignment quality was already very good. It can be seen that greater improvements were obtained for the simpler models.

As expected, the ME training takes a more important role when larger sizes of the corpus are used. For the smallest corpora, the number of training events for the ME models is very low, so it is not possible to disambiguate some translations/alignments for different contexts. For larger sizes of the corpora, greater improvements are obtained. Therefore, we expect to obtain better improvements when using even larger corpora.

After observing the common alignment errors, we plan to include more discriminant features that would provide greater improvements. We also expect improvements by performing a refined modeling of the rare/infrequent words, which are currently not taken into account by current the ME models.

7 Conclusions

In this paper, we have presented an efficient and straightforward integration of ME context-dependent models within a maximum likelihood training of statistical alignment models.

We have evaluated the quality of the alignments obtained with this new training scheme comparing the results with the baseline results. As can be seen in Section 6, we have obtained better alignment quality using the context-dependent lexicon models.

In the future, we plan to include more features in the ME model, such as dependencies with other source and target words, POS tags and syntactic con-

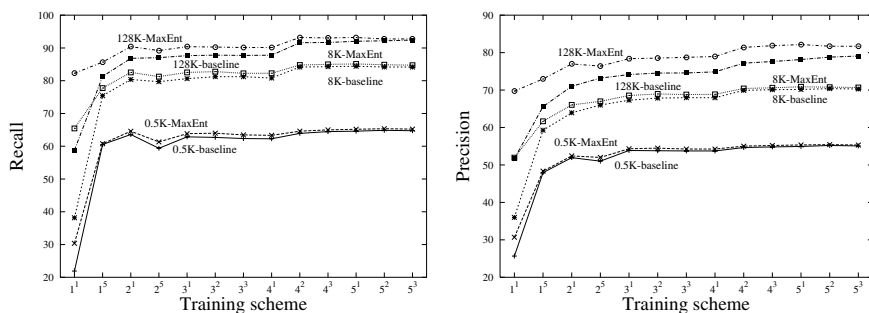


Fig. 2. Recall and Precision [%] results for Hansards task for different corpus sizes, for every iteration of the translation scheme.

stituents. We also plan to design ME alignment and fertility models. This will allow for an easy integration of more dependencies, such as second-order alignment models without running into the problem of an unmanageable number of alignment parameters. We have just started to perform experiments for a very distant pair of languages as is Chinese–English with very promising results.

References

1. Berger, A.L., Della Pietra, S.A., Della Pietra, V.J.: A maximum entropy approach to natural language processing. *Computational Linguistics* **22** (1996) 39–72 54, 57
2. Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* **19** (1993) 263–311 54, 55, 55, 55, 56, 56, 59
3. Darroch, J., Ratcliff, D.: Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics* **43** (1972) 95–144 55
4. Della Pietra, S.A., Della Pietra, V.J., Lafferty, J.: Inducing features in random fields. *IEEE Trans. on PAMI* **19** (1997) 380–393 57
5. Foster, G.: Incorporating position information into a maximum entropy/minimum divergence translation model. In: *Proc. of CoNLL-2000 and LLL-2000, Lisbon, Portugal* (2000) 37–52 54
6. García-Varea, I., Och, F.J., Ney, H., Casacuberta, F.: Refined lexicon models for statistical machine translation using a maximum entropy approach. In: *Proc. of the 39th Annual Meeting of the ACL, Toulouse, France* (2001) 204–211 54, 57
7. Och, F.J.: An efficient method for determining bilingual word classes. In: *9th Conf. of the Europ. Chapter of the ACL, Bergen, Norway* (1999) 71–76 57
8. Och, F.J., Ney, H.: Giza++: Training of statistical translation models (2001) <http://www-i6.informatik.rwth-aachen.de/~och/software/GIZA++.html>. 60
9. Och, F.J., Ney, H.: A comparison of alignment models for statistical machine translation. In: *COLING '00: The 18th Int. Conf. on Computational Linguistics, Saarbrücken, Germany* (2000) 1086–1090 60
10. Papineni, K., Roukos, S., Ward, R.: Maximum likelihood and discriminative training of direct translation models. In: *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*. (1998) 189–192 54
11. Rosenfeld, R.: A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language* **10** (1996) 187–228 54

Using Word Formation Rules to Extend MT Lexicons

Claudia Gdaniec and Esmé Manandise

IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598
{cgdaniec, esme}@us.ibm.com

Abstract. In the IBM LMT Machine Translation (MT) system, a built-in strategy provides lexical coverage of a particular subset of words that are not listed in its bilingual lexicons. The recognition and coding of these words and their transfer generation is based on a set of derivational morphological rules. A new utility extends unfound words of this type in an LMT-compatible format in an auxiliary bilingual lexical file to be subsequently merged into the core lexicons. What characterizes this approach is the use of morphological, semantic, and syntactic features for both analysis and transfer. The auxiliary lexical file (ALF) has to be revised before a merge into the core lexicons. This utility integrates a linguistics-based analysis and transfer rules with a corpus-based method of verifying or falsifying linguistic hypotheses against extensive document translation, which in addition yields statistics on frequencies of occurrence as well as local context.

1 Motivation

The LMT MT system [7], [8] has implemented a strategy [3] for recognizing a subset of word types with no entry in the bilingual core lexicons and for generating transfers based on rules of word formation for English. What characterizes the approach is the use of semantic and syntactic features for both analysis and transfer, a scoring system to assign levels of confidence to possible word structures, and the creation of transfers in the transformational component. This strategy improves parses based on correct part of speech (POS) and on context coding of unfound words. Furthermore, it generates transfers, which, if not perfect, convey the *gist* of the source word meaning. Returning such basic transfer contributes to the understandability of the text.

During text processing, the morphological analysis and transformation components provide lexical coverage of the unfound words. However, any subsequent encounter with the same input words is treated as if the words have not been encountered earlier. Thus, (i) morphological analysis and transformations apply more than once and (ii) there is no record of the unfound words encountered during text processing.

2 Purpose

This paper addresses points (i) and (ii) above in the light of the new approach implemented for the LMT MT system to automatically generate, during text processing, lists of bilingual lexicon entries that are not contained in the existing bilingual core lexicons. These entries are generated in the format of those lexicons and are associated with the morphological, syntactic, and semantic features that are important for LMT, which relies on the accuracy of such features for correct translation¹. Further, they need minimal human checking for linguistic correctness, which facilitates their later integration into the core lexicons. For the work described here, German was chosen as the source language and English as the target language.

3 Related Work

Most NLP systems rely on variable-size lexicons of base words. To improve application performance, various strategies in morphological analyzers (MA) have been implemented to provide lexical coverage of words with no corresponding entries in core lexicons [1], [6], [11]. The extent of the use of derivational, and to a lesser degree inflectional, morphology to handle unfound words seems to depend on the application. In MT, there are two difficulties in handling unfound words: Not only do the words need to be analyzed, but they also need acceptable transfers, which are difficult to generate for unfound words [3], [4], [9].

4 Strategy for Extending Lexicons

The acquisition of lexical items in LMT is based on a strategy for identifying words that are not found in the bilingual core lexicons but which can be related, by means of productive morphological rules, to some other word listed in the core lexicons [3]. This approach assumes the principle of compositionality for both the analysis and transfer of unfound words. In the process of acquiring linguistic knowledge about the source words, MA will assign, based on the affix list produced at an earlier stage in the morphological processing, a POS, syntactic arguments, and morpho-syntactic and semantic features. The rules will also create a bracketed word structure, which is passed on for later transfer generation, which itself is based on productive rules of word formation for the target language in the transformational component of LMT.

By creating the transfer structure in the transformations, which have access to syntactic and semantic features as well as to the source and transfer strings themselves, a certain degree of flexibility with respect to the form of the new transfer can be achieved. One new word or a whole new subtree can be created.

¹ We wish to thank Michael McCord and Sue Medeiros for making some additions to the LMT shell and the LMT transfer component in support of this work. Also, we wish to thank Michael McCord for comments on earlier versions of the paper.

To extend lexicons, an ALF can now be created during text processing to hold the source words with no corresponding lexical entries in the core lexicons together with their parts of speech, their morphological, syntactic, and semantic information along with the hypothesized translation(s). The linguistic information associated with each of the source words in ALF consists of the linguistic knowledge *accumulated* during morphological analysis, lexical look-up, and transformational processing; this knowledge may be quite complex and far from trivial.

Consider the following simple example of a generated LMT entry for a word in German² assuming it were not already in the lexicon:

- (1) Sprachpanscher
 < n (mor mf 2) st_professional st_device m
 > language waterer

The word would be found by means of data mining, which can be controlled for defined domains. The lexical lists are the result of combining rules during processing and unsupervised data mining. The word in (1) was analyzed on the basis of the German words *Sprache* and *panschen*.

5 Unsupervised Lexical Acquisition

5.1 The Process

Unsupervised lexical acquisition in the LMT MT system can be activated optionally by setting a flag at the beginning of text processing. It then creates ALF, which consists of unfound words in their uninflected form along with their corresponding postulated linguistic information and hypothesized transfer in LMT format.

To acquire as much lexical information as possible about unfound words, the system has been further enhanced to make unfound words inherit more than just the transfer word. The subcategorization of the base words from which they are derived is passed on as well.

The extract in (2) is from a German/English ALF. Some entries have accumulated linguistic knowledge quite similar to that which a lexicographer would enter.

- (2)
 Aachener
 < adj no_inflection no_adverb st_people city_name
 > Aachen
 abwaschbar
 < adj st_property deverbal no_adverb
 > washable

² The sign < indicates the beginning of the source analysis with part of speech, arguments, semantic and morpho-syntactic features. The sign > points to the transfer and any lexical transformations (xf) that apply to it.

Bewahrung

- < n obj (obj (pinf vor)(p vor dat)) (mor f 18) st_action
- > keeping

5.2 Delaying Writing to ALF

Although some of the information acquired through the morphological analysis could be written more easily to ALF while still in MA, there is evidence that it is premature to do so. Since there is no indication that output from MA will make it through and out of the parser and then into the transfer shell, it is better to write one and only one entry per word, i.e. the entry that was actually chosen by the parser, right or wrong. MA may create three analyses for one word. But only variants 1 and 2 may be verified by the corpus. Variant 1 may occur a hundred times, variant 2 may occur only five times. This information is relevant. Delaying writing to ALF until after the syntactic analysis gives this information and simultaneously verifies a hypothetical word from MA. German adjectives illustrate another reason for delaying. Adjectives and adverbs mostly look the same in German. In the lexicon, only one entry is listed in order to increase efficiency of parsing. MA generates an adjective. The syntactic analysis may choose the adverb function. After transformations, ALF contains the adjective entry together with information on its occurrence as an adverb (adv_use) in the actual document and an alternative adverb transfer. Consider the example below:

(3)

parodistisch

- < adj adv_use st_property
- > parodist (xf (adv_trans in a parodist way))

It may happen, however, that every single occurrence of this hypothesized adjective in a very large corpus is associated with this adverb use. In that case, one might want to reconsider the adjective POS.

5.3 Why Keep Multiple Instances of Unfound Words

The partial ALF list in (4) shows multiple identical German source entries.

(4)

Amtsrichterin

- < n nid title (p an dat)(mor f 19) human_indiv f
- > District Court Judge

Amtsrichterin

- < n nid title (p an dat)(mor f 19) human_indiv f
- > district court judge

Bonner

- < adj no_inflection no_adverb st_people city_name
- > Bonn

Bonner

```
< adj no_inflection no_adverb st_people city_name
> Bonn
```

Bonner

```
< n (mor mf 2) m city_name st_people
> Bonner
```

It might be argued that, if an unfound word like *Bonner* immediately above is going to be treated the same by the source analysis and the transformations, then it should be written to ALF only once. So if *Bonner* was encountered three times, one might think that it will be analyzed similarly three times. And one might argue that rather than undergoing the same analysis three times, the system should check what has already been written to the auxiliary file before going through the same motions **during text processing** a second, and a third time.

But notice that out of the three occurrences of *Bonner*, it is analysed as an adjective twice and as a noun only once. If this analysis is correct, it indicates the number of occurrences of the possible parts of speech generated by MA in actual documents. Further, the source word *Amtsrichterin* with the transfers *District Court Judge* and *district court judge* in (4) above provides information about the occurrence of the German word itself and its subcategorization (or optional/required slots). The translation in upper case means that the nid (noun identifier) slot is filled. This knowledge of course is made available through the lookup for the entry for the masculine form it is derived from. The subcategorizations are copied from the base word in the lexicon into the newly derived word. Multiple instances of an unfound word in ALF indicate the number of occurrences and the context of these words.

Ideally, the removal of duplicates, that is, of entries that are absolutely identical in the source description as well as in the transfer, should happen during text processing and data mining rather than after any type of processing. It would also be better if a tally was kept during processing of every true duplicate that is removed. Removing and keeping a count of duplicate occurrences is done at this moment in LMT after processing. The entries in (4) above will look like in (5). The number of occurrences for a particular part of speech appears to the left of the specific POS.

(5)

Amtsrichterin

```
< 2 n nid title (p an dat)(mor f 19) human_indiv f
> District Court Judge
> district court judge
```

Bonner

```
< 2 adj no_inflection no_adverb st_people city_name
> Bonn
< 1 n (mor mf 2) m st_people city_name
> Bonner
```

The treatment of unfound words where they are derived from existing listed bases in the core lexicon succeeds in returning different transfers whose difference is based on the argument structures of the base forms from which these unfounds are derived:

(6)

Beständigkeit

- < 2 n obj (p gegen) (mor f 18) st_property
- > resistantness
- > constantness

Beständigkeit is derived from the adjective *beständig* in the lexicon (where f means that this transfer is correct if a prepositional phrase with *gegen* is present):

(7)

beständig

- < adj (p gegen)
- > if f resistant
- if (gegen x) steady
- else constant

The slot structure from the base word is copied into that of the derived word. Then the transfer varies depending on whether the (p gegen) slot is filled or not. German nouns ending in *keit*, derived from adjectives, tend to have the semantic feature “property”. Equally, they tend to have a “carrier” of that “property”. This is realized when MA adds a slot for that carrier to the features of the derived noun. The beauty of the added object for nouns derived from adjectives is that this slot corresponds to the modified head of adjectives. The transfer test bearing on an adjective like *beständig* – where we get our transfer from – will fail or succeed based on the carrier of the property in the case of the derived noun. The validity of the correspondence of subcategorization is borne out in the following:

(8)

- | | | |
|----------------------------------|---|-------------------------------|
| a. Die Beständigkeit des Regens. | ! | The steadiness of the rain. |
| b. Der beständige Regen. | ! | The steady rain. |
| c. Die Beständigkeit der Furcht. | ! | The constantness of the fear. |
| d. Die beständige Furcht. | ! | The constant fear. |

There are advantages and disadvantages to making unfound words inherit the original subcategorization of the base words from which they are derived. So far, the evidence collected through text processing and data mining shows that (a) for words not derived from verbs, there seem to be only advantages and (b) for words derived from verbs to deverbal adjectives and nouns, including nominalized infinitives, there is a difficulty in determining the correct function of some of the derived arguments. Consider the German examples in (9) below.

(9)

- | | | |
|---|---|----------------|
| a. das Greifen (zur Zigarette) | ! | reaching (for) |
| b. das Greifen (der Saiten = object) | ! | capturing |
| c. das Greifen (der Maßnahme = subject) | ! | capturing |

Distinguishing between the object and the subject of a nominalized form is not straightforward. Consider the notorious ambiguity of *the shooting of the hunters*. If the verb is clearly intransitive, the genitive modifier can only be the subject. If both object and subject slots of the base verb were filled, the subject would be attached with a *durch* prepositional phrase. But if only one is filled, we are dealing with structural ambiguity that can be resolved only based on semantics or discourse context. If the verb is marked `human_agent` and `not_human_object`, and the genitive modifier is marked `human`, then it is also clear that it is the subject. If the verb is marked for `human_agent` and has no specification for the object, the preference for an encountered human genitive modifier is to consider it the subject, etc. In many cases, the successful application of transformations depends on the correct identification of arguments. More refined semantic analyses of the compatibility of the co-occurring words could help in this situation, but there is a limit to how deep these analyses can be.

5.4 Arguments against Unsupervised Lexicon Merge

ALF should not be used as an addendum lexicon at run-time while it is being created. It needs human revision before it can be considered a system lexicon because some classes of words need special handling. For example, deverbal words that would get a transfer of *not amendable* in ALF are often actually translated with a relative clause in transformations. A lexicon transfer would be acceptable if it were a good transfer, which can be difficult to compositionally create in these cases.

- (10) a document that can't be amended
a not amenable document

The foregoing is particularly pertinent for German nominalized infinitives. Transformations decide – based on various linguistic factors – where on the continuum of English nominalizations the translation takes place, from noun via gerund to infinitive or finite clause.

Further, in the LMT system, in order to be looked up at run-time, a word needs to be in a special, compiled form. Compiling the ALF every time a new word is added is not efficient considering that the percentage of those unfound words is relatively small compared with the total number of words in the document.

6 Results

The translation of 2 million words in a German daily newspaper, which covers many subject areas, has resulted in the creation of 10,600 entries in ALF. 46% of the words are true duplicates as defined under 5.3. Feminine forms of human individuals and titles like *friend*, *athlete*, *colleague* show many duplicates (61% of them). Deverbal nouns ending in *-ung* also have many duplicates (66% of them). Incorrect source analysis of POS is under 1%. Most of the incorrect analyses are words ending in *-er*;

they are incorrectly analyzed proper names or English loan words like *Springwater*, *Geißler*, *Ochsenzoller*. Inaccuracy of semantic and syntactic source features was not counted, but appears to be very low also.

Transfers for words with feminine endings tend to have very good transfers, which is not surprising since they are mostly identical with the masculine forms in English. Similarly, English *-ing* forms for German nominalized verbs tend to get mostly acceptable transfers, as do English *-able* adjectives.

The reason for unacceptable transfers often turns out to be less a problem of derivational morphology but rather a problem of compound analysis like *Romanleser* as Roman+leser versus Rom+anleser. Furthermore, some of the unacceptable transfers reflect inappropriate transfers of the original base words in the lexicon. Finally, spelling adjustments in target generation will be remedied with a call to the English target generation component of the LMT system from inside transformations that we are in the process of adding.

More work on target derivational morphology, which takes into account semantic information, will improve transfers like the suffixes *-ical* and *-ly* in *anarchical* versus *masterly*.

Not every entry in ALF reflects exactly what is found in the actual translation. ALF is written before LMT target generation is accomplished. The reason is that at this earlier stage the unfound source word is still seen as where and how it comes from MA and as one unit for the purpose of target synthesis. The possible discrepancy between ALF transfer and actual translation occurs for deverbal derivations, where the context of the transfer structure often decides whether a source word is translated as one unit or whether it is expanded into a whole clause. These transfers are marked with “or clausal transfer” in ALF.

The number of unfound derived words will stand in opposite relation to the number and salience of covered words in the lexicon. In a good lexicon that covers about 85,000 staple words of German, the number of derived unfound words will of course be lower than it would be for a smaller lexicon with less salient coverage.

(11) and (12) show some automatically generated output. The English transfers in (11) are quite acceptable while those in (12) are not.

(11)

Grenzörtchen

< n (mor nt 1) st_location st_diminutive

> small border place

Großmäuligkeit

< n obj (mor f 18) st_property

> loud-mouthedness

Grübeleien

< n obj (obj (p über)) (mor f 18) st_iterative human_agent deverbal

> musing

(12)

gipfelstürmerisch

< adj st_property

> summit forwardical

References

1. Byrd, R.J., Klavans, J.L., Aronoff, M., Anshen, F.: Computer Methods for Morphological Analysis. In Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics. (1986) 120-127
2. Daciuk, J.: Treatment of Unknown Words. In Proceedings of Workshop on Implementing Automata WIA'99, Berlin: Springer Verlag LNCS Series Volume 2214. (2001) 71-80
3. Gdaniec, C. Manandise, E., McCord, M.: Derivational Morphology to the Rescue: How It Can Help Resolve Unfound Words in MT. In Proceedings, MT Summit VIII, Santiago. CD Edition, compiled by John Hutchins. (2001)
4. Hutchins, W. J., Somers, H.L.: An Introduction to Machine Translation. London: Academic Press. (1992)
5. Jäppinen, H. Ylilammi, M.: Associative Model of Morphological Analysis: An Empirical Inquiry. Computational Linguistics 12(4). (1986) 257-272
6. Klavans, J.L. Jacquemin, C., Tzoukermann, E.: A Natural Language Approach to Multi-Word Term Conflation. In Proceedings, DELOS Workshop on Cross-Language Information Retrieval, ETHZ, Zurich. (1997)
7. McCord, M.C., Bernth, A.: The LMT Transformational System. In D. Farwell, L.Gerber, & E.Hovy (Eds.), Machine Translation and the Information Soup. Proceedings of the 3rd AMTA conference. Berlin: Springer. (1998) 344--354
8. McCord, M.C.: Slot Grammar: A system for simple construction of practical natural language grammars. In R. Studer (Ed.), Natural Language and Logic: International Scientific Symposium. Berlin: Springer. (1990) 118-145
9. McCord, M., Wolff, S.: The Lexicon and Morphology for LMT, IBM Research Division Research Report, RC 13403. (1988)
10. Sproat, R.W.: Morphology and Computation. Cambridge: MIT Press. (1992)
11. Woods, W.A.: Aggressive Morphology for Robust lexical Coverage. In Proceedings of the Sixth Applied Natural Language Processing Conference. (2000) 218-223

Example-Based Machine Translation via the Web

Nano Gough, Andy Way, and Mary Hearne

School of Computer Applications,
Dublin City University,
Dublin, Ireland.
`away@computing.dcu.ie`

Abstract. One of the limitations of translation memory systems is that the smallest translation units currently accessible are aligned sentential pairs. We propose an example-based machine translation system which uses a ‘phrasal lexicon’ in addition to the aligned sentences in its database. These phrases are extracted from the Penn Treebank using the Marker Hypothesis as a constraint on segmentation. They are then translated by three on-line machine translation (MT) systems, and a number of linguistic resources are automatically constructed which are used in the translation of new input.

We perform two experiments on testsets of sentences and noun phrases to demonstrate the effectiveness of our system. In so doing, we obtain insights into the strengths and weaknesses of the selected on-line MT systems. Finally, like many example-based machine translation systems, our approach also suffers from the problem of ‘boundary friction’. Where the quality of resulting translations is compromised as a result, we use a novel, *post hoc* validation procedure via the World Wide Web to correct imperfect translations prior to their being output to the user.

1 Introduction

Translation memory (TM) systems have rapidly become the most useful tool in the translator’s armoury. The widespread availability of alignment software has enabled the creation of large-scale aligned bilingual corpora which can be used to translate new, unseen input. Many people believe that existing translations contain better solutions to a wider range of translation problems than other available resources (cf. Macklovitch, 2000). However, the main problem with these knowledge sources is that they are aligned only at sentential level, so that the potential of TM systems is being vastly underused.

This constraint on what segments can be aligned is overcome in Example-based Machine Translation (EBMT) systems. Like TM systems, EBMT requires an aligned bilingual corpus as a prerequisite, but translational correspondences can in addition be derived at sub-sentential level, which is not possible in TM systems. Accordingly, EBMT systems *generate* translations of new input by combining chunks from many translation examples; the best that TM software can

do is to suggest the closest ‘fuzzy’ matches in its database for users to combine themselves in the formation of the translation.

In section 2, we show how the Marker Hypothesis (Green, 1979) can be used to create a ‘phrasal lexicon’ which renders the database of examples far more useful in building translations of new input. This lexicon was constructed by extracting over 200,000 phrases from the Penn Treebank and having them translated into French by three on-line machine translation (MT) systems. These three sets of translations were stored separately, and used as the basis for our EBMT system in translating two new testsets of NPs and sentences. As well as translating these examples using chunks from each of the individual sets of translations (A, B and C), in subsequent experiments we combined the three sets firstly in three new pairwise sets (AB, AC, BC), followed by combining them all together (ABC). The way that the chunks were combined and translations obtained is described in section 3. The results are presented in section 4.

Like many EBMT systems, our approach suffers from the problem of ‘boundary friction’. Where the quality of resulting translations is compromised as a result, we use a novel, *post hoc* validation procedure via the World Wide Web, described in section 5, to correct imperfect translations prior to their being output to the user. Finally, in section 6 we conclude and outline some ideas for further research.

2 The Phrasal Lexicon

Other researchers have also noted this advantage of EBMT systems over TM software, namely the ability to avail of sub-sentential alignment. Simard and Langlais (2001) propose the exploitation of TMs at a sub-sentential level, while Schäler *et al.* (2002) describe a vision in which a phrasal lexicon occupies a central place in a hybrid integrated translation environment.

Our phrasal lexicon was built in two phases. Firstly, a set of 218,697 English noun phrases and verb phrases was selected from the Penn Treebank. We identified all rules occurring 1000 or more times and then eliminated those that were not relevant, e.g. rules dealing only with digits. Of the rules with a RHS containing a single non-terminal, only those rules whose LHS is VP were retained in order to ensure that intransitive verbs were represented in our database of translations. In total, just 59 rules out of a total of over 29,000 were used in creating the lexicon.

These extracted English phrases were then translated using three different on-line MT systems:

- SDL International’s Enterprise Translation Server¹ (system A)
- Reverso by Softissimo² (B)
- Logomedia³ (C)

¹ <http://www.freetranslation.com>

² <http://trans.voila.fr>

³ <http://www.logomedia.net>

These MT systems were selected as they enable batch translation of large quantities of text. We found that the most efficient way to translate large amounts of data via on-line MT systems was to send each document as an HTML page where the phrases to be translated are encoded as an ordered list. The English phrases were therefore automatically tagged with HTML codes and passed to each translation system via the Unix ‘wget’ function. This function takes a URL as input and writes the corresponding HTML document to a file. If the URL takes the form of a query then the document retrieved is the result of the query, namely the translated web page. Once this is obtained, retrieving the French translations and associating them with their English source equivalents is trivial.

Despite the (often) poor output obtained from these systems, impressive results may still be obtained. We do not validate the translations prior to inserting them into our databases. Of course, if we were to do so, or use ‘better’ systems, then the results presented in section 4 would improve accordingly.

2.1 The Marker Lexicon

In their *Gaijin* system, Veale and Way (1997) propose the use of the Marker Hypothesis to create aligned chunks at sub-sentential level. The Marker Hypothesis is a psycholinguistic constraint on grammatical structure that is minimal and easy to apply. Given that it is also arguably universal, it is clear to see that it has obvious benefits in the area of translation.

The Marker Hypothesis states that all natural languages contain a closed set of specific lexemes and morphemes which indicate the grammatical structure of strings. As in *Gaijin*, we exploit such lists of known marker words for each language to indicate the start and end of segments. For English, our source language, we use the six sets of marker words in (1), with a similar set produced for French, our target language:

- (1) Det: {the, a, an, those, these, ...} Conj: {and, or, ...}
 Prep: {in, on, out, with, to, ...} Poss: {my, your, our, ...}
 Quant: {all, some, few, many, ...} Pron: {I, you, he, she, it, ...}

In a pre-processing stage, the aligned sentence pairs are traversed word by word, and whenever any such marker word is encountered, a new chunk is begun, with the first word labelled with its marker category (Det, Prep etc.). The following example illustrates the results of running the marker hypothesis over the phrase *on virtually all uses of asbestos*:

- (2) <PREP> on virtually, <QUANT> all uses, <PREP> of asbestos

In addition, each chunk must also contain at least one non-marker word, so that the phrase *out in the cold* will be viewed as one segment, rather than split into still smaller chunks.

For each $\langle \text{English}, \text{French}_X \rangle$ pair, where X is one of the sets of translations derived from separate MT systems (A, B, and C), we derive separate marker lexicons for each of the 218,697 source phrases and target translations. Given that English and French have essentially the same word order, these marker lexicons are predicated on the naïve yet effective assumption that marker-headed chunks in the source S map sequentially to their target equivalents T , i.e. $\text{chunk}_{S1} \rightarrow \text{chunk}_{T1}$, $\text{chunk}_{S2} \rightarrow \text{chunk}_{T2}, \dots, \text{chunk}_{Sn} \rightarrow \text{chunk}_{Tn}$. Using the previous example of *on virtually all uses of asbestos*, this gives us:

- (3) <PREP> on virtually : sur virtuellement
 <QUANT> all uses : tous usages
 <PREP> of asbestos : d'asbeste

In addition, we generalize over the phrasal marker lexicon along the lines of (Block, 2000). Taking (3) as input, we produce the templates in (4):

- (4) <PREP> virtually : <PREP> virtuellement
 <QUANT> uses : <QUANT> usages
 <PREP> asbestos : <PREP> asbeste

This allows other marker words of the same category to be substituted for those in the phrasal chunks. For instance, in our testset of NPs, we do not locate *the fully operational prototype*, the nearest approximation being *a fully operational prototype*. By replacing the marker word *a* with <DET>, we can search the generalized lexicon for the chunk <DET> *fully operational prototype*, retrieve its translation and insert translations for *the*. Errors of agreement in this insertion process may again be corrected using the techniques involved in section 5.

Finally, we take advantage of the further assumption that where a chunk contains just one marker word in both source and target, these words are translations of each other. Where a marker-headed pair contains just two words, therefore, we are able to extract a further bilingual dictionary. From the chunks in (3), we can extract the following six word-level alignments:

- (5) <PREP> on : sur <LEX> virtually : virtuellement
 <QUANT> all : tous <LEX> uses : usages
 <PREP> of : d' <LEX> asbestos : asbeste

That is, using the marker hypothesis method, smaller aligned segments can be extracted from the phrasal lexicon without recourse to any detailed parsing techniques. When matching the input to the corpus, we search for chunks in the order (original) phrasal dictionary $\rightarrow$ phrasal marker lexicon (cf. (3)) $\rightarrow$ generalized phrasal marker lexicon (cf. (4)) $\rightarrow$ word-level marker lexicon (cf. (5)), so that greater importance is attributed to longer chunks, as is usual in most EBMT systems. The word for word translation pairs are only used when a translation cannot be formed in any other way.

Given that verbs are not a closed class, we take advantage of the fact that the initial phrasal chunks correspond to rule RHSs. That is, for a rule in the Penn Treebank VP $\rightarrow$ VBG, NP, PP, we are certain (if the taggers have done

their job correctly) that the first word in each of the strings corresponding to this RHS is a VBG, i.e. a present participle. In such cases we also tag such words with the <LEX> tag, e.g. '<LEX> expanding : augmente'.

3 Chunk Retrieval and Translation Formation

In section 4, we describe two experiments, one on NPs and one on sentences. In this section, we describe the processes involved in retrieving appropriate chunks and forming translations for NPs only, these being easily extensible to sentences.

3.1 Segmenting the Input

In order to optimize the search process, a given NP is segmented into smaller chunks. The system then attempts to locate these chunks individually and to retrieve their relevant translation(s). We use an n -gram based segmentation method, in that all possible bigrams, trigrams and so on are located within the input string and subsequently searched for within the relevant knowledge sources.⁴

3.2 Retrieving Translation Chunks

We use translations retrieved from three different sources A, B and C. These translations are further broken down using the Marker Hypothesis, thus providing us with an additional three knowledge sources A', B' and C'—the phrasal marker lexicons. These knowledge sources can be combined in several different ways. We have produced translations using information from a single source (i.e. A/A', B/B' and C/C'), pairs of sources (i.e. A/A' & B/B' (=AB), A/A' & C/C' (=AC), and B/B' & C/C' (=BC)), and all available knowledge sources (i.e. A/A' & B/B' & C/C' (=ABC)). Each time a source language (SL) chunk is submitted for translation the appropriate target language (TL) chunks are retrieved and returned with a weight attached.

3.3 Calculation of Weights

We use a maximum of six knowledge sources: firstly, three sets of translations (A, B and C) retrieved using each on-line MT system; and secondly, three sets of translations (A', B' and C') acquired by breaking down the translations retrieved at the initial stage using the Marker Hypothesis. Within each knowledge source, each translation is weighted according to the following formula:

⁴ Of course, given our segmentation method, many of these n -grams cannot be found, given that new chunks are placed in the marker lexicon when a marker word is found in a sentence. Taking the NP *the total at risk a year* as an example, chunks such as 'the total at risk a' or 'at risk a' cannot be located, as new chunks would be formed at each marker word, so the best that could be expected here might be to find the chunks <DET> *the total*, <PREP> *at risk*, <DET> *a year* and recombine their respective translations to form the target string. In ongoing work, we are continuing to eliminate all such n -grams which are impossible to find from the search process.

$$(6) \quad \text{weight} = \frac{\text{no. occurrences of the proposed translation}}{\text{total no. translations produced for SL phrase}}$$

For the SL phrase *the house*, assuming that *la maison* is found 8 times and *le domicile* is found twice, then $P(\textit{la maison} \mid \textit{the house}) = 8/10$ and $P(\textit{le domicile} \mid \textit{the house}) = 2/10$. Note that since each SL phrase will only have one proposed translation within each of the knowledge sources acquired at the initial stage, these translations will always have a weight of 1.

If we wish to consider only those translations produced using a single MT system (e.g. A and A'), then we add the weights of translations found in both knowledge sources and divide the weights of all proposed translations by 2. For the SL phrase *the house*, assuming $P(\textit{la maison} \mid \textit{the house}) = 5/10$ in knowledge source A and $P(\textit{la maison} \mid \textit{the house}) = 8/10$ in A', then $P(\textit{la maison} \mid \textit{the house}) = 13/20$ over both knowledge sources. Similarly, if we wish to consider translations produced by all three MT systems, then we add the weights of common translations and divide the weights of all proposed translations by 6.

When translations have been retrieved for each chunk of the input string, these translated phrases must then be combined to produce an output string. In order to calculate a ranking for each TL sentence produced, we multiply the weights of each chunk used in its construction, thus favouring translations formed via larger chunks. Where different derivations result in the same TL string, their weights are summed and the duplicate strings are removed.

4 Experiments and System Evaluation

4.1 Experiment 1: Sentences

The automatically generated testset comprised 100 sentences, with an average length of 8.5 words (min. 3 words, max. 18).⁵ The sentences were segmented using the n -gram approach outlined in section 3. Following the submission of these sentences to each of the knowledge sources, translations were produced for 92% of cases for systems A and C, and 90% for system B. The same 8 sentences fail to be translated by any of the systems (or combinations of systems) owing to a failure to locate a word within the word-level lexicon. For 48% of the successful cases, the translation was produced by combining chunks found in either the original phrasal lexicon or the phrasal marker lexicon. In 28% of cases, the translation was produced by locating single words in the word-level lexicon and inserting these into the translation at the correct position. The remaining 16% of translations were produced with recourse to the generalized marker lexicon.

⁵ The testset itself adversely affected the results derived from this experiment. Given the preference for on-line systems to process S-level expressions, third person plural dummy subjects were provided. As a consequence, the VPs in our phrasal lexicon are for the most part in this corresponding form also. The majority of subject NPs in our sentence testset are singular, which almost guaranteed a lower quality translation. Nevertheless, the results achieved are still reasonable and can easily be improved by adding new, relevant translation examples to the system database.

While coverage is important, the quality of translations produced is arguably more important. All translations produced were evaluated by two native speakers of French with respect to the following classification schema:

- Score 3: contains no syntactic errors and is intelligible;
- Score 2: contains (minor) syntactic errors and is intelligible;
- Score 1: contains major syntactic errors and is unintelligible.

The results obtained are shown in Table 1. For the majority of those translations assigned a score of 2, the verb was either in the incorrect form, or the agreement between noun and verb was incorrect. Most of these examples may be corrected using the *post hoc* validation procedure outlined in section 5.

Table 1. Quality of Translations obtained for Sentence Testset

System	Score 1	Score 2	Score 3
A	14.2%	51.2%	34.6%
B	8.9%	54.7%	36.4%
C	4.4%	59.1%	36.5%

Like many other data-driven approaches to translation, our EBMT system produces many translation candidates for the user’s perusal. Another important issue, therefore, is that the ‘correct’ translation be as near to the top among those translations in which the system has the most confidence (i.e. the ‘best’ translation). We discuss issues pertaining to combining chunks from different on-line systems in section 4.3. For the individual systems, however, in over 65% of cases the ‘correct’ translation was ranked first by the system, and in all cases the ‘correct’ translation was located in the top five-ranked translations.

4.2 Experiment 2: Noun Phrases

A second experiment employing a testset of 200 noun phrases was subsequently undertaken. Here the average NP length was 5.37 words (min. 3 words, max. 10). In 94% of cases (188 NPs), at least one translation was produced for either system A, B or C. On average, about 54% of translations are formed by combining chunks from the phrasal lexicon, about 9% are produced by searching the generalized chunks, and about 37% are generated by inserting single words from the word-level lexicon at the appropriate locations in phrasal chunks. The failure to produce a translation in 6% of cases was invariably due to the absence of a relevant template in the generalized marker lexicon.

The translated NPs were again evaluated using the scale outlined in the previous section. The results achieved are summarized in Table 2, and are somewhat more definitive than with the sentence testset summarized in Table 1. Our EBMT system works best with chunks derived from system C, Logomedia, with a clear 7% more translations with no errors, and only 2.6% of translations deemed unintelligible. System B again outperforms System A.

Table 2. Quality of Translations obtained for NP Testset

System	Score 1	Score 2	Score 3
A	11.9%	51.4%	36.7%
B	4.8%	53.8%	41.4%
C	2.6%	49%	48.4%

As was the case with sentences, our EBMT system produces many translation candidates for NPs. For instance, the NP *a plan for reducing debt over 20 years* receives 14 translations using chunks from system A, 10 via B and 5 via C. When we combine chunks from more than one system, this rises to 224 for ABC. For the individual systems, in almost all cases, the ‘correct’ translation was located within the top five ranked translations proposed by the system, and at worst in the top ten.

4.3 Extending the Experiments

We also examined the performance of our EBMT system on both testsets when it has access to chunks from more than one system. We performed four more experiments—three pairwise comparisons (AB, AC, BC) and one threefold (ABC).

With respect to coverage, all four combinations translated 92% of the sentence testset. For the NP testset, in the pairwise comparison, coverage ranged from 94% (AB) to 95.5% (both AC and BC), while ABC translated 96% of the NPs.

We also evaluated translation quality using the same 3-point scale. For sentences, we observed that chunks involving some combination of system C perform better (AC and BC both achieving 48.9% top score, compared with AB’s 47.2%), with ABC outperforming any of the pairwise systems (50% of translations scoring 3). On the NP testset, AC (62.8% top score) and BC (62.3%) both outperform AB (58%), while ABC scores 3 for 70.8% of NPs, with only 0.5% (i.e. one NP) regarded as unintelligible.

Regarding the relative location of the ‘correct’ translation for sentences, the ‘correct’ translation is to be found in the top ten-ranked translations in all permutations of combinations of chunks, with at least 97.3% found in the top five and 54% ranked first. For NPs, the ‘correct’ translation is to be found in the top five-ranked translation candidates in almost all cases.

5 Validation and Correction of Translations via the Web

A translation can only be formed in our system when the recombination of chunks causes the input NP to be matched exactly. Therefore, if all chunks are not retrieved then no translation is produced. When a translation cannot be produced by combining the existing chunks, the next phase is to check whether a translation can be formed by the insertion of single marker words into the target

string. Given the NP *the personal computers*, this can be segmented into three possible chunks: *the personal*, *personal computers* and *the personal computers*. The chunk *personal computers* is the only one retrieved in the phrasal lexicon of our system. As it does not match the input NP exactly, its translation does not qualify as a complete translation, of course. The system stores a list of marker words and their translations in the word-level marker lexicon. A weight derived from the method in (6) is attached to each translation. The system searches for marker words within the string and retrieves their translations. In this case, the marker word in the string is *the* and its translation can be one of *le*, *la*, *l'* or *les* depending on the context. The system simply attaches the translation with the highest weight to the existing chunk (*ordinateurs personnels*) to produce the translation *la ordinateurs personnels*. Of course, the problem of boundary friction is clearly visible here.

However, rather than output this wrong translation directly, we use a *post hoc* validation and (if required) correction process based on (Grefenstette 1999). Grefenstette shows that the Web can be used as a filter on translation quality simply by searching for competing translation candidates, and selecting the one which is found most often. Rather than search for competing candidates, we select the ‘best’ translation and have its morphological variants searched for on-line. In the example above, namely *the personal computers*, we search for *les ordinateurs personnels* versus the wrong alternatives *le/la/l’ordinateurs personnels*. Interestingly, using Altavista, and setting the search language to French, the correct form *les ordinateurs personnels* is uniquely preferred over the other alternatives, as it is found 980 times while the others are not found at all. In this case, this translation overrides the ‘best’ translation *la ordinateurs personnels* and is output as the final translation. This process shows that while the Web is large, despite the fact that it is unrepresentative and may be seen to contain what might be considered ‘poor quality’ data, it remains a resource which is of great use in evaluating translation candidates.

6 Conclusions and Further Work

We have presented an EBMT system based on the Marker Hypothesis which uses *post hoc* validation and correction via the Web. A set of over 218,000 NPs and VPs were extracted automatically from the Penn Treebank using just 59 of its 29,000 rules. These phrases were then translated automatically by three on-line MT systems. These translations gave rise to a number of automatically constructed linguistic resources: (i) the original $\langle \textit{source}, \textit{target} \rangle$ phrasal translation pairs; (ii) the phrasal marker lexicon; (iii) the generalized phrasal marker lexicon; and (iv) the word-level marker lexicon. When confronted with new input, these knowledge sources are searched in turn for matching chunks, and the target language chunks are combined to create translation candidates.

We presented two experiments which showed how the system fared when confronted with NPs and sentences. For the former, we translated 96% of the testset, with 71% of the 200 NPs being translated correctly, and 99.5% regarded

as acceptable. For our 100 sentences, we obtained translations in 92% of cases, with a completely correct translation obtained 50% of the time, and an acceptable translation in 96.8% of cases. Importantly, the ‘correct’ translation was to be found in almost all cases in the top five-ranked translation candidates output by our system. Prior to outputting the best-ranked translation candidate, its morphological variants are searched for via the Web in order to confirm it as the final output translation or to propose a corrected alternative.

A number of issues for further work present themselves. The decision to take all rules occurring 1000 or more times was completely arbitrary and it may be useful to include some of the less frequently occurring structures in our database. Similarly, it may be a good idea to extend our lexicon by including more entries using Penn-II rules where the RHS contains a single non-terminal.

Furthermore, the quality of the output was not taken into consideration when selecting the on-line MT systems from which all our system resources are derived, so that any results obtained may be further improved by selecting a ‘better’ MT system which permits batch processing.

Finally, we want to continue to improve the evaluation of our system, firstly by experimenting with larger datasets, and also by removing any notion of subjectivity by using automatic evaluation techniques.

In sum, we have demonstrated that using a ‘linguistics-lite’ approach based on the Marker Hypothesis, with a large number of phrases extracted automatically from a very small number of the rules in the Penn Treebank, many new reusable linguistic resources can be derived automatically which can be utilised in an EBMT system capable of translating new input with quite reasonable rates of success. We have also shown that the Web can be used to validate and correct candidate translations prior to their being output.

References

1. Block, H. U.: Example-Based Incremental Synchronous Interpretation. In Wahlster, W. (ed.) *VerbMobil: Foundations of Speech-to-Speech Translation*, Springer-Verlag, Berlin Heidelberg New York (2000) 411–417
2. Green, T.R.G.: The Necessity of Syntax Markers. Two experiments with artificial languages. *Journal of Verbal Learning and Behavior* **18** (1979) 481–496
3. Grefenstette, G.: The World Wide Web as a Resource for Example-Based Machine Translation tasks. In *Proceedings of the ASLIB Conference on Translating and the Computer* **21**, London (1999)
4. Macklovitch, E.: Two Types of Translation Memory. In *Proceedings of the ASLIB Conference on Translating and the Computer* **22**, London (2000)
5. Schäler, R., Carl, M., Way, A.: Example-Based Machine Translation in a Hybrid Integrated Environment. In Carl, M., Way, A. (eds.) *Recent Advances in Example-Based Machine Translation*, Kluwer Academic Publishers, Dordrecht, The Netherlands (in press) (2002)
6. Veale, T., Way, A.: Gaijin; A Bootstrapping, Template-Driven Approach to Example-Based MT. In *Proceedings of the Second International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria (1997) 239–244

Handling Translation Divergences: Combining Statistical and Symbolic Techniques in Generation-Heavy Machine Translation

Nizar Habash and Bonnie Dorr

Institute for Advanced Computer Studies

University of Maryland

College Park, MD 20740

{habash,bonnie}@umiacs.umd.edu

<http://umiacs.umd.edu/labs/CLIP>

Abstract. This paper describes a novel approach to handling translation divergences in a Generation-Heavy Hybrid Machine Translation (GHMT) system. The translation divergence problem is usually reserved for Transfer and Interlingual MT because it requires a large combination of complex lexical and structural mappings. A major requirement of these approaches is the accessibility of large amounts of explicit symmetric knowledge for both source and target languages. This limitation renders Transfer and Interlingual approaches ineffective in the face of structurally-divergent language pairs with asymmetric resources. GHMT addresses the more common form of this problem, source-poor/target-rich, by fully exploiting symbolic and statistical target-language resources. This non-interlingual non-transfer approach is accomplished by using target-language lexical semantics, categorial variations and subcategorization frames to overgenerate multiple lexico-structural variations from a target-glossed syntactic dependency of the source-language sentence. The symbolic overgeneration, which accounts for different possible translation divergences, is constrained by a statistical target-language model.

1 Introduction

In this paper, we describe a novel approach to handling translation divergences using the Generation-Heavy Hybrid Machine Translation (GHMT) model introduced in [8]. The translation divergence problem is usually reserved for Transfer and Interlingual MT because it requires a large combination of complex lexical and structural mappings. A major requirement of these approaches is the accessibility of large amounts of explicit symmetric knowledge for both the source language (SL) and the target language (TL). This limitation makes Transfer and Interlingua inapplicable approaches to structurally-divergent language pairs with asymmetric resources. GHMT is a non-interlingual non-transfer¹ approach

¹ Although lexical transfer occurs in GHMT, it is merely a direct translation of unary lexical items from SL to TL not a structural transfer of SL-TL pairs of constituents.

that addresses the more common form of this problem, source-poor/target-rich, by fully exploiting symbolic and statistical TL resources.

SLs are only expected to have a syntactic parser and a translation lexicon that maps SL words to TL bags of words. No transfer rules or complex interlingual representations are required. The approach depends on the existence of rich TL resources such as lexical semantics, categorial variations and subcategorization frames to overgenerate multiple lexico-structural variations from a target-glossed syntactic dependency of the SL sentence. The symbolic overgeneration, which accounts for different possible translation divergences, is constrained by a statistical TL model.

The work presented here focuses on the generation component of GHMT and its handling of translation divergences. The next section describes the range of divergence types covered in this work and discusses previous approaches to handling them in MT. Section 3 describes the components of the GHMT approach. Finally, Section 4 addresses the interaction between statistical and symbolic knowledge in the system through illustrative examples.

2 Background: Translation Divergences

A translation divergence occurs when the underlying concept or “gist” of a sentence is distributed over different words for different languages. For example, the notion of floating across a river is expressed as *float across a river* in English and *cross a river floating* (*atravesó el río flotando*) in Spanish [4]. An investigation done by [6] found that divergences occurred in approximately 1 out of every 3 sentences in the TREC El Norte Newspaper Corpus². In the next section, we describe translation divergence types before turning to alternative approaches to handling them.

2.1 Translation Divergence Types

While there are many ways to classify divergences, we present them here in terms of five specific divergence *types* that can take place alone or in combination with other types of translation divergences. Table 1 presents these divergence archetypes with Spanish-English examples.³

- *Categorial Divergence*: Categorial divergence involves a translation that uses different parts of speech.
- *Conflation*: Conflation involves the translation of two words using a single word that combines their meaning. In Spanish-English translation, this divergence type usually involves a single English verb being translated using a combination of a light verb⁴, and some other meaning-heavy unit such as a noun or a progressive manner verb.

² LDC catalog no. LDC2000T51, ISBN 1-58563-177-9, 2000.

³ The divergence categories are described in more detail in [6].

⁴ Semantically “light” verbs carry little or no specific meaning in their own right such as *give*, *do* or *have*.

Table 1. Translation Divergence Types

Divergence	Spanish		English	%
Categorical	X <i>tener hambre</i>	(X <i>have hunger</i>)	X <i>be hungry</i>	98%
	X <i>tener celos</i>	(X <i>have jealousy</i>)	X <i>be jealous</i>	
Conflational	X <i>dar puñaladas a Z</i>	(X <i>give stabs to Z</i>)	X <i>stab Z</i>	83%
	X <i>ir pasando</i>	(X <i>go passing</i>)	X <i>pass</i>	
Structural	X <i>entrar en Y</i>	(X <i>enter in Y</i>)	X <i>enter Y</i>	35%
	X <i>pedir un referendum</i>	(X <i>ask-for a referendum</i>)	X <i>ask for a referendum</i>	
Head Swapping	X <i>cruzar Y nadando</i>	(X <i>cross Y swimming</i>)	X <i>swim across Y</i>	8%
	X <i>entrar corriendo</i>	(X <i>enter running</i>)	X <i>run in</i>	
Thematic	X <i>gustar a Y</i>	(X <i>please to Y</i>)	Y <i>like X</i>	6%
	X <i>doler a Y</i>	(X <i>hurt to Y</i>)	Y <i>hurt from X</i>	

- *Structural Divergence*: A structural divergence involves the realization of incorporated arguments such as subject and object as obliques (i.e. headed by a preposition in a PP) or vice versa.
- *Head Swapping*: This divergence involves the demotion of the head verb and the promotion of one of its modifiers to head position. In other words, a permutation of semantically equivalent words is necessary to go from one language to the other. In Spanish, this divergence is typical in the translation of an English motion verb and a preposition as a directed motion verb and a progressive verb.
- *Thematic Divergence*: A thematic divergence occurs when the verb’s arguments switch syntactic argument roles from one language to another (i.e. subject becomes object and object becomes subject). The Spanish verbs *gustar* and *doler* are examples of this case.

The last column in Table 1 displays a percentage of occurrences of the specific divergence type, taken from the first 48 unique instances of Spanish-English divergences from the TREC El Norte corpus. Note that there is often overlap among the divergence types with the categorical divergence occurring almost every time there is any other type of divergence. An extreme example of divergence type cooccurrence is *Maria tiene gustos de políticos diferentes*, which can be translated as *different politicians please Maria*. There are four divergence types in this pair: categorical (*gusto_{noun}* to *please_{verb}*), conflational (*tener gusto* to *please*), thematic (*Maria* and *politicians* switch syntactic roles) and structural (*politican* is an oblique in Spanish but an argument in English). This highlights the need for a systematic approach to handling divergences that addresses all their different types and the interactions amongst them rather than addressing specific cases one at a time.

2.2 Handling Translation Divergences

Since translation divergences require a combination of lexical and structural manipulations, they are traditionally handled minimally through the use of transfer

rules [9,16]. A pure transfer approach is a brute force attempt to manually encode all translation divergences in a transfer lexicon [5]. Very large parsed and aligned bilingual corpora have also been used to automatically extract transfer rules [17,21]. This approach depends on the availability of such resources, which are very scarce.

Alternatively, more linguistically-sophisticated techniques that use lexical semantic knowledge to detect and handle divergences have been developed. One approach uses Jackendoff’s Lexical Semantic Structure (LCS) [10,11] as an interlingua [4]. LCS is a compositional abstraction with language-independent properties that transcend structural idiosyncrasies by providing a granularity of representation much finer than syntactic representation. LCS has been used in several projects such as UNITRAN [3] and ChinMT [20]. As an example, the Spanish sentence *Juan cruza el río nadando* can be “composed” as the following LCS using a Spanish LCS lexicon as part of the interlingual analysis:

```
(1) [event CAUSE JOHN
      [event GO JOHN [path ACROSS JOHN [position AT JOHN RIVER]]]
      [manner SWIM+INGLY]]
```

In the generation phase, this same LCS is “decomposed” using English LCS lexicon entries to yield *John swam across the river*.

Another approach enriches lexico-structural transfer at Mel’čuk’s Deep Syntactic Structure (DSyntS) level [18] with cross-linguistic lexical semantic features [19]. Transfer lexicon rules are written to capture generalizations across the language pair instead of addressing specific paired instances. As an example, the following transfer rule can be used to handle the head swapping divergence discussed in the last example.

```
(2) @TRANS_CORR
    @EN V1 [cat:verb manner:M]
        (ATTR Y [cat:prep path:P event:go] (II N))
    @SP V2 [cat:verb path:P event:go]
        (II N ATTR Z [manner:M])
```

Here, a transfer correspondence is established between the different components of two DSyntS templates. Note how the manner variable M and the path variable P switch dominance.

A major limitation of these interlingual and transfer approaches (whether using lexical semantics or corpus-based) is that they require a large amount of explicit symmetric knowledge for both SL and TL. We propose an alternative approach called Generation-Heavy Machine Translation approach (GHMT). This approach is closely related to the hybrid approach described in [12,13,14]. The idea is to combine symbolic and statistical knowledge in generation through a two step process: (1) Symbolic Overgeneration followed by (2) Statistical Extraction. The hybrid approach has been used previously for generation from semantic representations [13] or from shallow unlabeled dependencies [1]. GHMT extends on this earlier work by including structural and categorial expansion of SL syntactic dependencies as part of the symbolic overgeneration component. The fact that

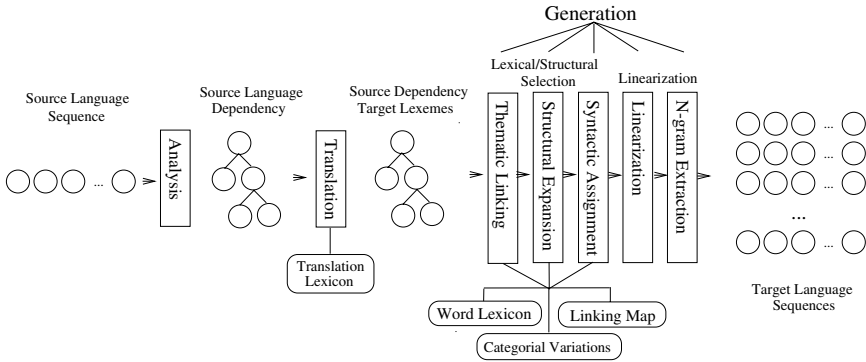


Fig. 1. Generation-Heavy Machine Translation

GHMT doesn't require semantically analyzed SL representations or structural transfer lexicons, makes it perfect for handling translation divergences with relatively minimal lexical resources on for the SL. The overgeneration is constrained by linguistically-motivated rules that utilize TL lexical semantics and is independent of the SL preferences. The generated lexico-structural combinations are then ranked by the statistical extraction component⁵. Figure 1 presents an overview of the complete MT system.

3 Generation-Heavy Machine Translation

The three phases of GHMT—Analysis, Translation and Generation—are very similar to other paradigms of MT: Analysis-Transfer-Generation or Analysis-Interlingua-Generation [5]. However, these phases are not symmetric. Analysis relies only on the SL sentence parsing and is independent of the TL. The output of Analysis is a deep syntactic dependency that normalizes over syntactic phenomena such as passivization and morphological expressions of tense, number, etc. Translation converts the SL lexemes into bags of TL lexemes. The dependency structure of the SL is maintained. The last phase, Generation, is where most of the work is done to manipulate the input lexically and structurally and produce TL sequences. Next we will describe the generation resources followed by an explanation of the generation sub-modules.

3.1 Generation Resources

The generation component utilizes three major TL resources (see Figure 1). First, the word-class lexicon defines verbs and prepositions in terms of their sub-

⁵ Another non-interlingual non-transfer approach is Shake-and-Bake MT [2] which overgenerates TL sentences and symbolically constraints the output by "parsing it." It differs from GHMT in two ways: (1) Shake-and-Bake is a purely symbolic approach and (2) it requires symmetric resources for SL and TL.

categorization frames and lexical conceptual primitives. A single verb or preposition can have multiple entries for each of its senses. For example, among other entries, *run*₁ as in (*John*_{agent} *ran*_{cause-go_{identificational}} *store*_{theme}) is distinguished from *run*₂ as in (*John*_{theme} *ran*_{go_{locational}}). Second, the categorial-variation lexicon relates words to their categorial variants. For example, *hungerv_V*, *hunger_N* and *hungry_{AJ}* are clustered together. So are *cross_V* and *across_P*; and *stab_V* and *stab_N*. Finally, the syntactic-thematic linking map relates syntactic relations (such as subject and object) and prepositions to the thematic roles they can assign. For example, while a subject can take on just about any thematic role, an indirect object is typically a *goal*, *source* or *benefactor*. Prepositions can be more specific. For example, *toward* typically marks a *location* or a *goal*, but never a *source*.

3.2 Generation Sub-modules

The generation component contains five steps (the five rightmost rectangles shown earlier in Figure 1). The first three are responsible for lexical and structural selection and the last two are responsible for linearization.

Initially, the SL syntactic dependency—now containing TL lexemes—is converted into a thematic dependency. The syntax-thematic linking is achieved through the use of thematic grids associated with English (verbal) head nodes together with the syntactic-thematic linking map. This step is a *loose* linking step that does not enforce the subcategorization-frame ordering or preposition specification. This looseness is important for linking from non-English subcategorization frames. For example, although the sentence **Mary filled water in the glass* is a bad English sentence (albeit good Korean), its arguments are mapped correctly as *agent*, *theme* and *location* respectively. The correct mapping is reached because *in* is a *location*-specifying preposition.

The next step is structural expansion, which explores conflated and head-swapped variations of the thematic dependency. Conflation is handled by examining all verb-argument pairs (V_{head}, Arg) for *conflatability*. For example, in *John put salt on the butter*, *to put salt on* can be conflated as *to salt* but *to put on butter* cannot be conflated into *to butter*. The thematic relation between the argument and its head together with other lexical semantic features constrain this structural expansion. Head Swapping is restricted through a similar process that examines head-modifier pairs for *swappability*.

The third step turns the thematic dependency into a full TL syntactic dependency. Syntactic positions are assigned to thematic roles using the verb class subcategorization frames and argument category specifications.

These three steps in the generation component address different translation divergence types. The thematic linking normalizes the input with respect to the thematic and structural divergences. Once the thematic roles are identified and surface syntactic cases are invisible, structural expansion can take place to handle conflation and head-swapping possibilities. The very common categorial divergence is handled at the structural expansion step too, but it is also fully addressed in the syntactic assignment step.

Finally, in the linearization step, a rule based grammar is used to create a word lattice that encodes the different possible realizations of the sentence. The grammar is implemented using the linearization engine oxyGen [7]. Sentences are ranked with Nitrogen’s Statistical Extractor using a uni/bigram model of two years of Wall Street Journal [14].

4 Discussion: Symbolic-Statistical Knowledge and Translation Divergences

A preliminary evaluation of GHMT conducted by [8] found that four of every five Spanish-English divergences can be generated using structural expansion and categorial variations⁶. Here we look at the interaction between symbolic and statistical knowledge⁷ in GHMT in the context of divergence handling using the following two illustrative Spanish-English divergent examples:

(3) Yo le di puñaladas a Juan.

```
I stabbed John .    [ LENGTH 4, SCORE 0.670270 ]
I gave a stab at John .    [ LENGTH 7, SCORE -2.175831 ]
I gave the stab at John .    [ LENGTH 7, SCORE -3.969686 ]
I gave an stab at John .    [ LENGTH 7, SCORE -4.489933 ]
I gave a stab by John .    [ LENGTH 7, SCORE -4.803054 ]
I gave a stab to John .    [ LENGTH 7, SCORE -5.045810 ]
I gave a stab into John .    [ LENGTH 7, SCORE -5.810673 ]
I gave a stab through John .    [ LENGTH 7, SCORE -5.836419 ]
I gave a knife wound by John .    [ LENGTH 8, SCORE -6.041891 ]
I gave John a knife wound .    [ LENGTH 7, SCORE -6.212851 ]
```

(4) Juan tiene hambre.

```
John is hungry .    [ LENGTH 4, SCORE -7.111878 ]
John hunger .    [ LENGTH 3, SCORE -7.474780 ]
John is starved .    [ LENGTH 4, SCORE -7.786015 ]
John is a hunger .    [ LENGTH 5, SCORE -8.173432 ]
John has a hunger .    [ LENGTH 5, SCORE -8.613148 ]
John is a famine .    [ LENGTH 5, SCORE -8.666368 ]
John is the hunger .    [ LENGTH 5, SCORE -8.829170 ]
John is the famine .    [ LENGTH 5, SCORE -8.871368 ]
John be hungry .    [ LENGTH 4, SCORE -9.038840 ]
John is a starvation .    [ LENGTH 5, SCORE -9.105497 ]
```

In both examples, the system generates several valid English translations expressing a wide range of linguistic phenomena such as conflation and the dative alternation. This is accomplished purely by GHMT’s TL resources without any

⁶ The rest of the cases require more conceptual knowledge, pragmatic knowledge and/or hard-wiring of idiomatic non-decompositional expressions.

⁷ For a general discussion of the value of statistical knowledge in hybrid systems, see [15].

specification of or linking to the SL structures *dar puñaladas* or *tener hambre*. The most correct form of the output is ranked highest in both cases: *I stabbed John* and *John is hungry*. However, the ranking of the other choices doesn't reflect fluency or accuracy well. For example, *I gave John a knife wound* ranks much lower than *I gave an stab at John* although the former is more fluent. And the generation of *John is a hunger* as a variant of *John is hungry* is an inaccurate translation. These issues can be traced back to either the symbolic component's overgeneration or the statistical component's *under*-extraction.

One case highlighting the issue of fluency is the generation of the sequence *John hunger* in example (4). Here, the symbolic rules are not enforcing any subject-verb agreement, which results in allowing the sequence *John hunger* in the generated word lattice together with *John hungers*. However, the statistical model fails to rank *John hunger* lower than *John hungers*, which doesn't even make it to the top-ten sequences. This failure is likely due to the smoothing model used for handling unseen bigrams, which depends on the word unigrams instead. *Hunger* is a more common unigram than *hungers*.

Another case relevant to the fluency issue is the underspecification of preposition selection for the verb *give* in example (3). The current constraint is that the selected preposition *could* assign the thematic role, in this case, *goal*. Thus, the preposition *by* selected for *I gave a stab by John* has the locational not the agentive sense. The statistical model failure here is likely due to uni/bigrams enforcing fluency locally on a very small window. A possible solution on the statistical side is to use structural n-gram language models (similar to [1]) to capture long-distance dependencies between the verb and its modifiers.

The case of generating *John is a hunger* in example (4) reflects the dependency of GHMT on TL statistical knowledge as opposed to translingual knowledge of translation divergences. The argument here is that generating the metaphoric *John is a hunger* is a "compromise" of accuracy worth taking when generated with more likely sequences such as *John is hungry*. If the SL input was a metaphoric *John BE hunger*, then other verbs besides *be* would not be generated to start with and the smaller search space will allow the less likely metaphoric expression to be selected. This argument is, of course, hard to evaluate—and in example (4), *John is a hunger* ranks higher than the poetic *John has a hunger*. This ordering is a result of the statistical extraction use of bigrams in our current system which picks *John is a X* over *John has a X* regardless of *X*.

5 Conclusions and Future Work

We have described how translation divergences are handled in a novel hybrid machine translation approach, GHMT, that transcends the need for symmetry of resources required by Transfer and Interlingual approaches. This is accomplished by exploiting symbolic and statistical TL resources such as lexical semantics and statistical language models. The interaction between the symbolic and statistical components in GHMT is open for further research. Proposed modifications to

these components include stricter symbolic rules to limit extraneous overgeneration and structural language models to improve statistical extraction. Both of these modifications make use of TL knowledge and resources *only*, which is consistent with GHMT's generation-heavy philosophy.

Our immediate future work will involve an expansion of the linearization grammar to handle large-scale Spanish-English GHMT. Moreover, we plan to conduct a more extensive evaluation of the behavior of the system as a whole including a comparative analysis of other models of Spanish-English MT (an interlingual model and a transfer model). And finally, we are interested in testing SL-independence by retargeting the system to Chinese input.

Acknowledgments

This work has been supported, in part, by ONR MURI Contract FCPO.810548265 and Mitre Contract 010418-7712. We would like to thank Irma Amenero, Clara Cabezas, and Lisa Pearl for their help collecting and translating the Spanish data. We would also like to thank Amy Weinberg for helpful conversations.

References

1. Bangalore, S., Rambow, O.: Exploiting a Probabilistic Hierarchical Model for Generation. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000). Saarbrücken, Germany (2000) 87, 91
2. Beaven, J.L.: Shake-and-Bake Machine Translation. In: Proceedings of the 14th International Conference on Computational Linguistics (COLING-92). Nantes, France (1992) 88
3. Dorr, B.J.: Interlingual Machine Translation: A Parameterized Approach. *Artificial Intelligence*. Vol. 63 (1993) 429–492 87
4. Dorr, B.J.: Machine Translation: A View from the Lexicon. The MIT Press, Cambridge, MA (1993) 85, 87
5. Dorr, B.J., Jordan, P.W., Benoit, J.W.: A Survey of Current Research in Machine Translation. In: Zerkowicz, M. (ed.): *Advances in Computers*, Vol. 49. Academic Press, London (1999) 1–68 87, 88
6. Dorr, B.J., Pearl, L., Hwa, R., Habash, N.: DUSTer: A Method for Unraveling Cross-Language Divergences for Statistical Word-Level Alignment. In: Proceedings of the Fifth Conference of the Association for Machine Translation in the Americas (AMTA-2002). Tiburon, California (2002) 85, 85
7. Habash, N.: oxyGen: A Language Independent Linearization Engine. In: Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas (AMTA-2000). Cuernavaca, Mexico (2000) 90
8. Habash, N.: Generation-Heavy Hybrid Machine Translation. In: Proceedings of the International Natural Language Generation Conference (INLG-02). New York (2002) 84, 90
9. Han, H.C., Lavoie, B., Palmer, M., Rambow, O., Kittredge, R., Korelsky, T., Kim, N., Kim, M.: Handling Structural Divergences and Recovering Dropped Arguments in a Korean/English Machine Translation System. In: Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas (AMTA-2000). Cuernavaca, Mexico (2000) 87

10. Jackendoff, R.: *Semantics and Cognition*. The MIT Press, Cambridge, MA (1983) 87
11. Jackendoff, R.: *Semantic Structures*. The MIT Press, Cambridge, MA (1990) 87
12. Knight, K., Hatzivassiloglou, V.: Two-Level, Many-Paths Generation. In: *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*. Cambridge, MA (1995) 252-260 87
13. Langkilde, I., Knight, K.: *Generating Word Lattices from Abstract Meaning Representation*. Technical Report, Information Science Institute, University of Southern California (1998) 87, 87
14. Langkilde, I., Knight, K.: Generation that Exploits Corpus-Based Statistical Knowledge. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics joint with the 17th International Conference on Computational Linguistics (ACL/COLING-98)*. Montreal, Canada (1998) 704-710 87, 90
15. Langkilde, I., Knight, K.: The Practical Value of N-Grams in Generation. In: *Proceedings of the International Natural Language Generation Workshop* (1998) 90
16. Lavoie, B., Kittredge, R., Korelsky, T., Rambow, O.: A Framework for MT and Multilingual NLG Systems Based on Uniform Lexico-Structural Processing. In: *Proceedings of the 1st Annual North American Association of Computational Linguistics (ANLP/NAACL-2000)*. Seattle, WA (2000) 87
17. Lavoie, B., White, M., Korelsky, T.: Inducing Lexico-Structural Transfer Rules from Parsed Bi-texts. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics – DDMT Workshop*. Toulouse, France (2001) 87
18. Mel'čuk, I.: *Dependency Syntax: Theory and Practice*. State University of New York Press, New York (1988) 87
19. Nasr, A., Rambow, O., Palmer, M., Rosenzweig, J.: Enriching Lexical Transfer With Cross-Linguistic Semantic Features (or How to Do Interlingua without Interlingua). In: *Proceedings of the 2nd International Workshop on Interlingua*. San Diego, California (1997) 87
20. Traum, D., Habash, N.: Generation from Lexical Conceptual Structures. In: *Proceedings of the Workshop on Applied Interlinguas, North American Association of Computational Linguistics/Applied Natural Language Processing Conference (NAACL/ANLP-2000)*. Seattle, WA (2000) 34-41 87
21. Watanabe, H., Kurohashi, S., Aramaki, E.: Finding Structural Correspondences from Bilingual Parsed Corpus for Corpus-based Translation. In: *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*. Saarbrücken, Germany (2000) 87

Korean-Chinese Machine Translation Based on Verb Patterns¹

Changhyun Kim, Munpyo Hong, Yinxia Huang, Young Kil Kim, Sung Il Yang,
Young Ae Seo, and Sung-Kwon Choi

NLP Team, Human Information Processing Department, Electronics and Telecommunications Research Institute, 161 Gajeong-Dong, Yuseong-Gu Daejeon, 305-350, Korea
{chkim, hmp63108, yinxia, kimyk, siyang, yaseo, choisk}@etri.re.kr

Abstract. This paper describes our ongoing project “Korean-Chinese Machine Translation System”. The main knowledge of our system is verb patterns. Each verb can have several meanings and each meaning of a verb is represented by a verb pattern. A verb pattern consists of a source language pattern part for the analysis and the corresponding target language pattern part for the generation. Each pattern part, according to the degree of generality, contains lexical or semantic information for the arguments or adjuncts of each verb meaning. In this approach, accurate analysis can directly lead to natural and correct generation. Furthermore as the transfer mainly depends upon verb patterns, the translation rate is expected to go higher, as the size of verb pattern grows larger.

1 Introduction

Machine Translation (henceforth MT) requires correct analysis of source languages, appropriate generation into target languages and a large amount of knowledge such as rules, statistics or patterns. Developing an MT system is in many cases labor-intensive and very costly, therefore effective knowledge acquisition and management and incremental improvement of the quality according to the amount of knowledge accumulation are the keys to the success of MT.

A rule based method suffers from knowledge acquisition problems and the difficulties of the consistent management. A statistical method has difficulties in connecting the previous statistical knowledge with new knowledge. It is not easy to reflect linguistic phenomena and peculiarities directly into knowledge, either. Patterns have several formats such as sentence-based patterns [1], [6], phrase-based patterns [8] and collocation-based patterns [2], [7]. Sentence-based patterns use a whole sentence as a template and transfer the input sentence in one step. However, it suffers mainly from data sparseness. Phrase-based patterns can be employed for both analysis and transfer. The transfer takes place on the phrasal level. Collocation-based patterns are used for lexical transfer, in other words, the transfer unit is a word. In [6] a Japanese-English

¹ This work was funded by the MIC (Ministry of Information and Communication) of Korean government in the framework of the “CAT System based on TM for 4 Asian Languages Project”

prototype MT system with a hybrid engine was introduced. The authors claim that they can take full advantage of EBMT systems in their framework whilst an RBMT system works for the parts that are not matched to the input sentence. This approach still requires certain amount of electrically available bilingual corpus which is not the case for the Korean and Chinese language pair. And also it suffers from the same problems that an RBMT system usually shows. To add a new language into the framework can also be costly.

During the development of Korean-English MT System from 1999 to 2001 we experienced that the linguistic peculiarities between two languages can be well captured in the verb pattern-based framework and that the translational divergences could be directly reflected in the knowledge. The system performance also improved as the number of verb patterns increased: [3], [5]

In the verb pattern-based framework each meaning of a polysemous verb is represented by a verb pattern. A verb pattern consists of a source language pattern part for analysis and the corresponding target language part for generation. Each pattern part, according to the degree of generality, contains lexical or semantic information for the arguments or adjuncts of each verb meaning. In this approach, accurate analysis can directly lead to natural and correct generation. The overall performance of a system is expected to be enhanced in proportion to the number of verb patterns. With respect to the reusability of knowledge, verb patterns for Korean-English MT could be recycled in many cases by just replacing the English pattern part with Chinese counterparts, thus saving the cost and time for knowledge construction. Among 80,000 Korean-English verb patterns, about 60,000 patterns could be reused for Korean-Chinese patterns. The rest 20,000 were Korean-English specific patterns.

In the following sections we elaborate about Korean-Chinese MT System being developed in the "CAT System based on TM for 4 Asian Languages Project" launched in 2001 under the auspices of MIC of Korean government. The system consists of a Korean morphological analyzer, a verb-pattern based parser and a generator consisting of a verb phrase linker and a word generator. The morphological analyzer first readjusts a sentence into appropriate morphological units, performs morphological analysis and finally ranks the results using statistical information. The parser readjusts the results of the morphological analyzer into syntactic units and analyses dependency relation using verb patterns. If the parser fails to find the matching verb patterns, it tries passive/causative rule application, extension of the meaning of arguments (or adjuncts) of verb patterns, or pseudo-verb pattern matching. A verb pattern, however, describes only the structure between a verb and its arguments (or adjuncts) not between verbs in a sentence. For this reason the parser analyses predicate-predicate structure employing statistical information of verb endings. The generator determines the Chinese translation for connectives and arranges the order of each connective clause. Words not covered by verb patterns are generated and positioned also in this stage.

2 Verb Pattern

Phrase-based patterns can be used for both syntactic analysis and transfer: [8]. The term ‘verb pattern’ we adopt is to be understood as a kind of subcategorization frame of a predicate. However, a verb pattern in our approach is slightly different from a subcategorization frame in the traditional sense. The main difference between a verb pattern and a subcategorization frame is that a verb pattern is always linked to the target language word (predicate of the target language). Therefore a verb pattern is employed not only in the analysis but also in the transfer phase so that the accurate analysis can directly lead to the natural and correct generation. In the theoretical linguistics a subcategorization frame always contains arguments of a predicate. An adjunct of a predicate or a modifier of an argument is usually not included in it. However, for the purpose of MT, these words must be taken into account. In translation adjuncts of a verb or modifiers of an argument can seriously affect the selection of target words, as can be seen in the following example:

Korean : 이(this) 달도(month) 다(completely) 갔다(to pass by)

English : This month is up

Verb Pattern: A=시간(time)!이 다(completely):b 가(to pass by)!다:PAST > A be:v up :: 가다 2²

In this example, the more appropriate and natural translation of the Korean verb ‘가다 (to pass by)’ is ‘be up’, if modified by the adverb ‘다 (completely)’. This kind of conflation divergence can be easily handled in the pattern-based approach by directly encoding the words in the pattern. Verb patterns simply annotate an adverb with a marker (b) and link the adverb and the verb to a conflated English expression. Idiomatic usages of a verb can also be treated easily within verb patterns. A frozen argument in an idiomatic expression is just to be lexicalized with an appropriate postposition separated by “!”. For example, a Korean idiomatic expression “호감이 가다 (to be favorably disposed toward someone)” can be described as in the following:

A=사람!가 B=사람!에게 호감이 가!다 > A be:v favorably disposed toward B:OBJ :: 가다 3

The noun ‘호감 (a favorable impression)’ is not overtly expressed in the target language expression. If an expression in the source language side is not marked, it will not be considered any more in the further phase of the translation. Postpositions such

² In ‘A=시간!이’, ‘A’ is a variable, ‘시간’(time) is a semantic code, ‘이’ is a postposition representing nominative case, ‘!’ is a separator between a content word and a functional word. In ‘가!다:PAST’, ‘가’ is the lexical form, ‘다’ is a functional word and a condition is described after ‘:’. ‘>’ is the separator between source language pattern and target language pattern. Comments appear after ‘::’ and ‘가다 2’ means the 2nd verb pattern of the verb ‘가다’

such as ‘에 게(eykey)’, ‘가(ka)’, are normalized Korean postpositions which correspond to syntactic case markers or postpositions.

A Korean verb pattern is linked to its corresponding Chinese verb pattern by the symbol ‘>’. The arguments in the left-hand side of a verb pattern are basically represented with semantic features such as ‘시간 (time)’, ‘공간 (location)’, ‘교통 (transportation)’, etc. About 200 hierarchical semantic features are currently made use of to cover the semantic information of nouns. The right-hand side of ‘>’ is the corresponding target word expression. In the current stage of the development we have about 60,000 Korean-Chinese verb patterns and we expect to have about 200,000 patterns by the end of the year.

3 Parser

The parser first pre-processes the results of morphological analysis into syntactic units, and then detects the local dependency structure for each predicate using verb patterns. If the parser fails to find the appropriate verb patterns, it tries passive/causative rule application, extension of the meaning of arguments (or adjuncts) of verb patterns, or pseudo-verb pattern matching. Although some adverbs are dealt with in our verb patterns, they are not exhaustively covered in the verb patterns. In such cases the parser computes the scopes of adverbs using statistical information. Finally the parser determines the structure between predicates employing predicate-predicate structure patterns. In this section, we focus on analyzing the predicate-argument-adjunct structures and predicate-predicate structures.

3.1 Predicate-Argument-Adjunct Structure Analysis

A dependency structure is employed in predicate-argument-adjunct analysis. As described previously, verb patterns describe not only arguments but also adjuncts. Each verb on a dependency tree is compared with verb patterns. In case of a prenominal clause, the modifree of the clause can be an argument or adjunct of the modified predicate. So the prenominal clause including the modifree is compared together with verb patterns. When more than one verb patterns are found, each verb pattern is evaluated according to the below criteria :

- the number of matched arguments
- the number of mismatched arguments
- the locality of verb patterns

Verb patterns with more matched arguments and fewer mismatched arguments score higher points³. Also, the verb patterns with shorter distance are preferred to those with longer distance⁴.

³ Our parser adds or subtracts some scores according to the degree of preference and selects the highest scored tree as the best tree.

In a pattern-based approach, the coverage problem is one of the highly debated topics. We cope with this issue in three ways: passive/causative rule application, extension of the meaning of arguments (or adjuncts) of verb patterns, or pseudo-verb pattern matching.

In an agglutinative language like Korean derivational rules for causativization and passivization which alter the number of arguments of a verb, are highly productive. To treat such cases a special device is needed, for otherwise it will require manual construction of the verb patterns. Instead of constructing passivized- and causativized verb patterns manually, we employ rules to convert verb patterns into corresponding passivized- or causativized verb patterns half-automatically. This was made possible because of the regularities in transforming syntactic cases to and from active/passive, active/causative forms as the following table shows:

Table 1. Transformation Rule between Active, Passive, Causative Forms

Passive → Active	Causative → Active
① for transitive verbs	① for verb/adjective
② subject → object	② subject → adverb
③ adverb → subject	③ object,adverb(에 게) → object,subject
④subject,adverb(에 서)→adverb(로), object(를)	④ adverb(에 게) → subject
	⑤ object → subject

Below is an example before and after applying the above rules.

Before : 보다 (to see) : A=사람(human)!가 B= 제품(product)!를 보(see)!다 > A 看:v B
After : 보이다 (to be seen) : A=사람(human)!에 게 B= 제품(product)!가 보이(seen)!다 > B 被 A 看:v

The arguments of a verb pattern are represented as lexical forms or semantic codes. Lexical forms have to be considered as such, but semantic codes are to be extended reasonably within a hierarchical structure. For example, if there exists a verb pattern “A=human eat B=edible”, then the semantic code ‘human’ can be extended to ‘animal’ intuitively without any problem. To this end we classified verbs into ‘[+human]’, ‘[+animal]’ according to their subjecthood. For example, ‘뛰다 (to run)’ is ‘[+animal]’ and thus the argument ‘A=사람(human)!가’ can be extended to ‘A=동물(animal)!가’.

뛰다 (to run) : A=사람(human)!가 B=장소(place)!로 뛰(run)!다 > A 跑:v 向 B

If the parser fails to find the exact matching verb patterns up to this point, it finally tries pseudo-verb pattern matching. Pseudo-verb patterns of a verb are the verb pat-

⁴ In a sentence ‘A B C D’, for example, the verb pattern matched with arguments A, B has shorter distance than the one with arguments A, C.

terns of the verb with the most similar characteristics in their argument distributions. For example, the verb patterns of ‘좋아하다(to like)’ is very similar to the verb patterns of ‘사랑하다(to love)’ in their argument distributions. For each verb we compute 5 most similar pseudo-verbs beforehand and use them in parsing time.

Topic markers such as ‘는 (nun)’, ‘도(to)’ and postposition ellipsis can be interpreted as several syntactic cases and thus cause difficulties in syntactic case resolution.⁵ The current parser restricts the interpretation only to nominative and accusative cases and applies the same method to both phenomena. The modifiee of a prenominal clause also falls under the case of postposition ellipsis. To deal with topic markers and postposition ellipsis, the parser applies the nominative postposition ‘가 (ka)’ and the accusative postposition ‘를 (lul)’ in turn to topic markers and postposition ellipsis and compares them with verb patterns. If the meanings between nouns are the same and the case is still empty then the case resolution is succeeded.

3.2 Predicate-Predicate Structure Analysis

Predicate-predicate structure analysis determines the structure between predicates in a sentence. For example, the Korean sentence “그[he]-는[topic] 자신[himself]-이[subj] 범인[criminal]-이[subj] 아니[to be not]-라고[connective ending] 밝히[to declare]-고[connective ending] 잠적했 [to disappear]-다 [final ending] (He disappeared after declaring that he was not the criminal)” has 3 predicates “아니 (to be not)”, “밝히 (to declare)” and “잠적하 (to disappear)” and can have 2 candidate structures as in Fig. 1⁶.

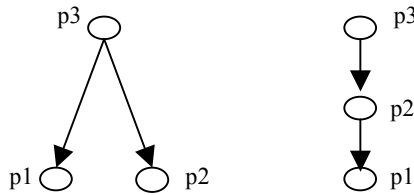


Fig. 1. Predicate Structure Candidates for a Sentence with 3 Predicates

To determine the hierarchical structure between predicates, the semantic relationship between them needs to be explored, inevitably requiring rich semantic information, which is very difficult to obtain in the current situation. For this reason we only make use of statistical information between connective endings. In Korean, postpositions and verb-endings are well developed and are possible to represent the meanings of sentences [4]. Connective endings combine two events denoted by two predicates with rhetoric relations such as cause, reason, expectation, or condition. So, we can assume that connective predicate endings can play the role of weak semantic relation information. Basically a predicate-predicate structure pattern describes the depend-

⁵ ‘nun’ and ‘to’ respectively correspond ‘wa’ and ‘mo’ in Japanese.

⁶ In Korean, the preceding words cannot be the heads of the following words. So, p1 is the 1st predicate ‘아니 (to be not)’, p2 the 2nd ‘밝히 (to declare)’, p3 the 3rd ‘잠적하 (to disappear)’

ency structure of predicates by their lexical connective endings with frequency information. But the semantic relations denoted by predicate connective endings are too simple and we are planning to adopt the semantic classification of predicates and the semantic relations between predicates. Theoretically any number of predicates in a predicate-predicate pattern is possible from 2 to n, but we are currently considering only 2 and 3 predicate patterns from the practical point of view. For each dependency tree, preference value of each predicate-predicate structure candidate is computed.

4 Generator

The first step in the generation is to construct the structure between predicates in a sentence. And then each arguments or adjuncts are generated for each predicate. Verb phrase linker is the component for the first part and verb phrase generator is responsible for the second part.

4.1 Verb Phrase Linker

Due to the similarity of the order of predicates in complex sentences between Korean and Chinese, Chinese sentences can be generated in many cases by just combining the translated Chinese verb phrases using appropriate conjunction words. The verb phrase linker uses verb phrase link patterns to link verb phrases. Below is the basic structure of a verb phrase link pattern :

VP1[어서] VP2[다] > VP1[ECONJ:[croot := [所以]]] VP2

“VP1[어서] VP2[다]” means that two verb phrases (VP1, VP2) are linked with a connective ending ‘어서’. ‘ECONJ’ is the Chinese conjunction corresponding to ‘어서’. The verb phrase linker traverses a dependency tree to detect dependency relations between verb phrases and produces generation information using verb phrase link patterns.

4.2 Verb Phrase Generator

The verb phrase generator translates Korean verb phrases into Chinese. This can be divided into three steps: verb phrase generation, noun phrase generation and adverb phrase generation. If a verb phrase is exactly matched to a verb pattern, its translation will be guided by the verb pattern. In case no exact matching verb pattern is found for a verb phrase, co-occurrence patterns are consulted. The co-occurrence patterns have a quadruple format, “(semantic feature or lexemes, functional word, verb, frequency of co-occurrences)”. For example, “(장소[place], 에[to], 가[to go], 12)” shows that a postposition, “에” and the semantic feature “장소 [place]” appeared with a verb,

“가 [to go]” 12 times in verb patterns. The meaning of the words that have no correspondence in a verb pattern is inferred from the most frequent co-occurrence entries.

Noun phrase generation uses both rules and patterns. Noun phrase generation in Chinese seems to be much simpler than in English. For example, a particle ‘의(of)’, used for adnominal noun phrase, is translated into “的” only, in comparison with various Korean-English translations.

Adverb generation is more complex than noun phrase generation. Contrary to Korean, the position of Chinese adverbs depends on various factors:

- i. classes of adverbs,
- ii. semantic relation between predicates and adverbs,
- iii. whether the predicate is a single character or not,
- iv. whether the mood of a sentence is imperative or not.

Adverb generation is principally based on rules. For example, when an adverb is a locative one with no directionality, it is located between the subject and the predicate. If it does have directionality, it is positioned after the predicate. But, the analysis of the semantic relation between the predicate and adverb is not so easy and it is one of the main topics to tackle this year.

5. Evaluation

At the end of the first year of the development we evaluated the Korean-Chinese MT-system. The test suite is made up of 100 sentences randomly extracted from primary school textbooks. The average length of the test sentence is approximately 10.5 words. A Korean-Chinese bilingual gave scores to the sentences according to the following criteria:

Score	Criteria
4	the meaning of the sentence is preserved
3	the meaning of the sentence is partially preserved (the predicate of the sentence is correctly translated, so that the skeletal meaning of the sentence is preserved, however some arguments or adjuncts of the sentence are not correctly translated)
2	at least one phrase is correctly translated
1	at least one word is correctly translated
0	no output

The translation rate was calculated by the following equation:

$$\text{Translation Rate} = \text{Summation of the Scores} / 4$$

The translation rate calculated was 44.5 %. Considering that we only have constructed 30 % of the verb patterns we aim to have by the end of this year (i.e. 60,000

of 200,000) the rate was not disappointing at all. However, because we are well aware of the fact that the translation rate does not always reflect the matching rate of the verb patterns, we decided to see how much was the matching rate of the verb patterns, which is our main concern in this approach.

In the 100 test sentences we found 151 verbs and predicative adjectives. 23.18 % of 151 predicates were successfully translated with verb patterns. For about 45 % of the predicates no verb-pattern was found. In the rest 31.82 % a verb pattern was matched to the input predicates, however, the verb-patterns contained wrong Chinese translations.⁷ Our conclusion from this is that when we have verb patterns in high quality, we can expect the translation rate to rise at least up to 55%. As the size of the verb-pattern DB grows larger, and the Korean analysis technique improves much more, we expect that the translation rate can go above 55 %. Other problems that did not fall under the verb-pattern category are as follows

Korean Sentence : 나는 동생이 신을 신도록 도와 줍니다.

I helped my younger brother to put on his shoes.

Generated Output : 我弟弟穿鞋帮助。

After Correction : 我帮助弟弟穿鞋。

In generation we usually maintain the relative order of two verb phrases in a Korean sentence as in the verb phrase linker. But, it causes problems when verb phrases are in special syntagmatic relations as in the above example.

In the case of a compound ‘방 청소 (room cleaning)’, ‘청소 (cleaning)’ plays the role of a predicate and ‘방 (room)’ is the theme of the action. The generator has to consider such cases and translate them as predicate-argument structure in Chinese.

Korean Sentence : 나는 오빠와 함께 방 청소를 합니다.

I clean the room with my elder brother.

Generated Output : 我与哥哥一起房间打扫。

After Correction : 我与哥哥一起打扫房间。

6 Conclusion

This paper described our ongoing project “Korean-Chinese MT System”. A verb pattern consists of a source language pattern part for analysis and the corresponding target language pattern part for generation. Each pattern part, according to the degree of generality, contains lexical or semantic information for the arguments or adjuncts of a verb meaning. In this approach, accurate analysis can directly lead to natural and correct generation. Furthermore as the transfer is mainly dependent upon the verb

⁷ The poor quality of some verb-patterns was in part due to the fact that the verb-patterns constructed by a few university students did not undergo the reexamination procedure by Korean-Chinese bilinguals in the first year. The reexamination of the verb-patterns is in progress.

patterns, the translation rate is expected to rise, as the size of verb pattern grows larger. In the second phase of the project we are planning to improve Korean analysis module and to construct more verb patterns half-automatically. By the end of this year we expect to have constructed about 200,000 verb patterns and to achieve 55% translation rate.

References

1. Hiroyuki, K., Kida, Y., Morimoto, Y. : Learning Translation Templates From Bilingual Text, in Proceeding of the 15th International Conference on Computational Linguistics, Nantes, France (1992) 678-678
2. McTait, K., Trujillo, A. : A Language-Neutral Sparse Data Algorithm for Extracting Translation Patterns, in Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation, England (1999) 98-108
3. Kim, Y.K., Seo, Y.A., Choi, S.K., Park, S.K. : Estimation of Feasibility of Sentence Pattern-Based Method for Korean to English Translation System, Int'l Conference on Computer Processing of Oriental Languages (2001)
4. Nam, K.S., Ko, Y.G. : The Standard Theory of Korean Grammar, Top Publication (1993)
5. Seo, Y.A., Kim, Y.K., Seo, K.J., Choi, S.K. : Korean to English Machine Translation System based on Verb phrase: CaptionEye/KE, Proceedings of the 14th KIPS Fall Conference (2000)
6. Shirai, S., Bond, F., Takahashi, Y. : A Hybrid Rule and Example-based Method for Machine Translation, Proceedings of NLPRS'97 (1997) 49-54
7. Smadja, F., McKeown, K., Hatzivassiloglou, V. : Translating Collocations for Bilingual Lexicons: A Statistical Approach, in Computational Linguistics (1996), Vol.21(4) 1-38
8. Hideo, W. : A Method for Extracting Translation Patterns from Translation Patterns. In Proceedings of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation, Kyoto, Japan (1993) 292-301

Merging Example-Based and Statistical Machine Translation: An Experiment

Philippe Langlais and Michel Simard

Laboratoire de Recherche Appliquée en Linguistique Informatique (RALI)
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
C.P. 6128, succursale Centre-ville
H3C 3J7, Montréal, Québec, Canada
<http://www.rali-iro.umontreal.ca>

Abstract. Despite the exciting work accomplished over the past decade in the field of Statistical Machine Translation (SMT), we are still far from the point of being able to say that machine translation fully meets the needs of real-life users. In a previous study [6], we have shown how a SMT engine could benefit from terminological resources, especially when translating texts very different from those used to train the system. In the present paper, we discuss the opening of SMT to examples automatically extracted from a Translation Memory (TM). We report results on a fair-sized translation task using the database of a commercial bilingual concordancer.

1 Introduction

The past decade witnessed exciting work in the field of Statistical Machine Translation. We are still far however from the point of being able to say that machine translation fully meets the needs of real-life users and it is a well known fact that translators remain reluctant to post-edit the output of a machine translator (statistical or not).

The present work is largely inspired by two studies we have previously conducted. In a first one [6], we investigated how a statistical engine behaves when translating a very domain-specific text far different from the corpus used to train both the translation and language models used by the engine. We measured a significant drop in performances mainly due to *out-of-vocabulary* (unknown) words and specific terminology that the models handle poorly. We proposed to overcome the problem by providing the engine with available (non statistical) terminological resources.

In a second study [13], we investigated how a database of past translations could help a human translator in his work. Such *Translation Memories* (TM) already exist, but typically operate on complete sentences, thus limiting their usefulness. We showed in our study that an impressive coverage of the source-language (SL) text could be obtained (up to 95%) by systematically querying the memory with sub-sentential sequences of words of the text to translate,

from which we may automatically retrieve useful target-language (TL) material. Other works in the same vein reported comparable encouraging results [3].

In the present study, we extend these lines of work by feeding a statistical translation engine with examples automatically extracted from the database of an online bilingual concordancer: TSRALI¹. This concordancer allows a user to query a large collection of French-English bitexts (more than 100 million words per language), aligned at the level of sentences. A full description is given in [8].

The approach we investigate here involves three main steps which are described in the following sections. The first consists in chunking the SL text to be translated; each identified chunk is then submitted to the TM (Section 2). The second step extracts from all the TL material returned by a query the portions that are likely to be useful for a translation (Section 3). In a third step (Section 4), these pieces of TL text are fed to a statistical engine in order to produce a translation. In Section 5, we report on an experiment we conducted made on a fair-sized corpus. We conclude with a discussion in Section 6.

2 Looking up SL Sequences in tsrali

In this paper, we call *Translation Memory* (TM) a database of existing translations. Conceptually, it can be viewed as a collection of pairs $\langle S, T \rangle$, where S is a SL segment of text, T is a TL segment, and S and T are translations of one another. In our case, S and T are typically single sentences, although in some cases S or T may be empty (“untranslated sentences”) or consist of a short sequence of sentences (anywhere between 2 and 5). We call these pairs *couples*.

Using standard full-text indexation techniques, it is possible to efficiently extract from such a collection all couples that contain some given sequence of word-forms in one language or the other. This is precisely what TSRALI is designed to do. Given a SL sentence $S = s_1 \dots s_m$, our plan is to use such a TM to propose TL translations for partial sequences s_i^j of S .

In previous work using a similar setup [13], we established that concentrating on *syntactically motivated* sequences of S was more productive than looking up all possible sequences. To identify these sequences in S , we employ a *chunker*, i.e. a system that identifies basic syntactic constituents.

Our chunker essentially follows the lines of [11]: it relies on a part-of-speech tagger (in our case, a hidden Markov model rather than a maximum entropy tagger), and proceeds in successive tagging stages, each working on the previous stages’ output. The first stage is a standard POS tagger: it associates a POS tag p_i to each word-token of s_i of S . The second stage takes as input a symbol obtained by combining s_i and p_i , and outputs so-called *IOB* tags c_i . These tags take one of the forms: *B-X*: first word of a chunk of type X ; *I-X*: non-initial word in a X chunk; *O*: word outside of any chunk.

The last and final stage is designed to provide the chunker with more context: it takes as input a symbol obtained by combining POS and IOB tags p_i, c_i, p_{i+1}

¹ <http://www.tsrali.com>.

and c_{i+1} , and produces “revised” IOB tags c'_i on the output. An example of the resulting bracketing is shown in Figure 1.

$[_{NP}$ The government] $[_{VP}$ is putting] $[_{NP}$ a \$2.2 billion tax] $[_{PP}$ on]
 $[_{NP}$ Canada] $[_{NP}$'s most vulnerable industry] , $[_{NP}$ the airline industry] .

Fig. 1. Output of the chunker.

We then search our TM for all sequences that begin and end at chunk boundaries (sequences of O tags are viewed as chunks in this process). We also exclude search sequences of less than two words, and sequences exclusively made up of very frequent word-forms (we use a stop-list of the 20 most frequent SL words). Figure 2 shows the matching sequences for the example of Figure 1, and Figure 3 shows a sample matching couple for one of the sequences found in the TM.

The government / The government is putting / is putting / a \$2.2 billion tax /
a \$2.2 billion tax on / a \$2.2 billion tax on Canada / Canada / Canada 's most vulnerable
industry / 's most vulnerable industry / , the airline industry / the airline industry

Fig. 2. The 10 sequences found in the translation memory.

Source: Yes , **the airline industry** is an important industry .
Target: Oui , l' industrie aérienne est un secteur important .

Fig. 3. Sample match for sequence “, *the airline industry*”.

3 Identifying Potentially Useful TL Units

For each SL sequence s_i^j of S , we extract a (possibly empty) set of couples from the TM. In order to come up with translation proposals for the sequence s_i^j , from each of these couples $\langle S_k, T_k \rangle$, we must now identify the part of T_k that translates the initial sequence. We make the simplifying assumption that this translation will itself be a sequence of words from T_k (no discontinuous translations).

For this task, we use a sequence alignment method that recursively segments the SL and TL text, each time choosing the segmentation that maximizes an association score between the matched pairs of segments. This scoring function approximates $P(t_k^l | s_i^j)$, the probability of observing the TL sequence t_k^l , given the SL sequence s_i^j :

$$Score(i, j+1, k, l+1) = \delta(j-i | l-k) \prod_{K=k}^l \sum_{I=i}^j \frac{tr(t_K | s_I)}{j-i} \quad (1)$$

where the $tr(t|s)$ are the lexical parameters of a statistical translation model (IBM model 1 [2]) and $\delta(m|n)$ represents the probability of observing a sequence of m words as the translation of a sequence of n words. In practice, we also make the simplifying assumption that the δ distribution is uniform over “reasonable” values of m .

Given a pair of sequences $\langle s_i^{j-1}, t_k^{l-1} \rangle$, the alignment procedure finds optimal segmentation points I and K , and the best way of pairing up the resulting sub-sequences (in *parallel*, or in *reverse*):

$$\langle I, K, d \rangle = \operatorname{argmax}_{I, K, d} \begin{cases} \operatorname{Score}(i, I, k, K) \times \operatorname{Score}(I, j, K, l) & (d = \textit{parallel}) \\ \operatorname{Score}(i, I, K, l) \times \operatorname{Score}(I, j, k, K) & (d = \textit{reverse}) \end{cases}$$

It then proceeds recursively on pairs of sequences $\langle s_i^{I-1}, t_k^{K-1} \rangle$ and $\langle s_I^j, t_K^l \rangle$ (or $\langle s_i^{I-1}, t_K^l \rangle$ and $\langle s_I^j, t_k^{K-1} \rangle$ if $d = \textit{reverse}$).

We have found that we can both improve alignment results and significantly reduce the search-space for this procedure by forcing it to consider only “syntactically motivated” segmentation points I and K . To do this, we first run the SL and TL segments of each couple through text chunkers, identical to the one described in Section 2. We then consider as valid only those segmentation points that lie at chunk boundaries. Figure 5 shows sample alignments for two different SL sentences.

4 Merging Example-Based and Statistical MT

4.1 The Translation Engine

We have extended the decoder (statistical machine translator) of [10] to a trigram language model. The basic idea of this search algorithm is to expand hypotheses along the positions of the TL string while progressively covering the SL ones. Every TL word may be associated with any l adjacent SL words; the decoder thus accounts for the notion of *fertility* [2], even if IBM model 2 does not incorporate this notion.

The decoder is a dynamic programming scheme based on a recursion which results from straight manipulations of the following maximization equation, where the SL sentence to translate is s_1^J , and l indicates the fertility of the TL word t_i :

$$\hat{t}_1^J = \max_I \left[\underbrace{p(J|I)}_{\text{length}} \max_{t_1^I} \prod_{i=1}^I \left[\underbrace{p(t_i|t_{i-2}t_{i-1})}_{\text{trigram}} \max_{j,l} \prod_{\bar{j}=j-l+1}^j \left\{ \underbrace{p(i|\bar{j}, J, I)}_{\text{alignment}} \underbrace{p(s_{\bar{j}}|t_i)}_{\text{transfer}} \right\} \right] \right] \quad (2)$$

We refer the reader to [10] for the formal description of the recursion, and instead give in Figure 4 a sketch of how a translation is built-up.

4.2 Feeding the TL Examples to the Decoder

There are many possible strategies to integrate the contributions of the TM into the decoding process. One of these is to “reward” the decoder whenever it

Input: $s_1 \dots s_j \dots s_J$

```

for all TL position  $i = 1, 2, \dots, I_{max}$  do
  for all valid hypothesis at stage  $i - 1$  do
    for all TL word  $t_i$  do
      for all free SL position  $j$  do
        for all fertility  $l$  do
          Consider  $t_i$  to be the translation of the  $l$  SL words  $s_j^{j+l-1}$ 

```

Fig. 4. Sketch of our decoder

generates hypotheses that contain part or all of a TL example sequence. However, because of the various pruning strategies used to keep the decoding time reasonable, even highly promising TL hypotheses may never be examined.

Instead, we investigated an approach that rests on the assumption that among the TL sequences extracted from the TM, there must be at least one which corresponds to a valid (usable) translation of the associated SL sequence.

In this perspective, the task of the statistical engine is to discover which TM sequences are most likely to be useful, as well as to determine (by optimizing equation 2 over the full sentence) the most likely target positions of these sequences. Hence what we are looking for is the most likely sentence that contains one TL sequence from the TM per matched SL sequence.

In the extreme case, if in the sentence of Figure 1, the only sequence submitted to the TM was *the airline industry*, with only one association returned *l'industrie du transport aérien*, our search algorithm would end up with a translation which contains only this French passage; the position of this sequence in the final translation being a by-product of the maximization operation.

5 Experiment

5.1 Practical Details

The TSRALI database used as TM for extracting sub-sentential translation proposals contains all the debates of the Canadian Parliament (the Hansard), published between April 1986 and December 2001, in all over 100 million words in each language. The French and English documents were aligned at the sentence level using *SFIAL*, a somewhat improved implementation of the method proposed in [12].

The English chunker used for sub-sentential extraction and sequence alignment was composed of three distinct HMM-based taggers ([5]), as discussed in Section 2. All taggers were trained on data from the Penn Treebank, more specifically the training set provided for the CONLL 2000 shared task. Its performance is essentially similar to that of [11].

The architecture of the French chunker used in the sequence alignment procedure is similar to that of the English tagger. The first stage HMM (POS tagger) was trained on a 160 000-word hand-tagged portion of the Hansard. The second

and third stage HMM's were trained on a portion of the *Corfrans* corpus [1], a collection of articles from the French newspaper *Le Monde*, manually annotated for syntax. The overall performance of the French chunker is much worse than the English (we estimate around 70% precision and recall). This is likely attributable to the small size of the training corpus, less than 1 500 sentences in all, compared to over 5 000 sentences for the English chunker.

To train our statistical translation engine, we assembled a bitext composed of 1 639 250 automatically aligned pairs of sentences. In this experiment, all tokens were folded to lower case before training. The inverted translation model (French-to-English) we used in equation 2 is essentially an IBM model 2. The language model is an interpolated trigram trained on the English sentences of our bitext.

The test corpus for our experiments comes from recent transcripts of the Hansard (March 2002), from which we extracted a passage of 1260 sentences, with an average SL length of 19.4 words. For the purposes of this experiment, English was taken as the source language and the French Hansard translation served as the "oracle".

5.2 Example-Based Sequences

More than 22 000 queries were successfully submitted to the TSRALI TM; the average length of successful queries was 4.6 words (the longest one was 17-word long). Of the 1 260 sentences, 12% did not generate any successful query, and less than 4% were found verbatim in the TM, which reinforces the claim that sentence-based TM systems are only useful in very specific tasks (revisions of previously translated documents, very repetitive sub-domains, etc.). We excluded these sentences from our test corpus. Successful queries produced more than 1.2 million TL examples, for an average of 56 examples per query.

In [13], we proposed a coarse evaluation of this example extraction process by assuming a user who tries to produce the oracle translation by juxtaposing pieces of the proposed TL examples. Clearly, a system that proposes a multitude of TL examples is more likely to cover the oracle translation, but at the cost of an increased burden for the user. We therefore evaluated this process in terms of *precision* (the quantity of useful TL material proposed) and *recall* (the proportion of the oracle translation covered by the proposed material). These figures were computed under various user-scenarios.

One somewhat unrealistic scenario assumed that the user constructed his translation by cutting and pasting freely (even single words) from the proposed TL examples. This corresponds to the ratios reported in the first line of Table 1. If we only allowed the user to paste entire TL examples, as proposed by our system, the ratios dropped by more than half (line 3 of Table 1). Line 2 in that table is an in-between scenario where we allow the user to grab sequences of at least two words from the TL proposals².

² All these ratio have been measured after applying the *cover* filter described in [13].

user-scenario	precision	recall	f-measure
cut&paste 1	50.5%	36.6%	42.4%
cut&paste 2	20.8%	27.4%	23.6%
paste-only	14.5%	20.0%	16.8%

Table 1. Results of using TM examples to assist a human translator. The f-measure is the harmonic average of precision and recall.

What these results indicate is that, when the user is only allowed to juxtapose entire pieces of the proposed TL material, one out of every 7 TL sequences retrieved from the TM is useful to produce 20% of the oracle translation.

5.3 Translating

At the time of writing, we have only translated SL sentences that contain at most 30 words (actually more than 90% of the sentences of the test corpus). We tested our translation engine with and without the addition of the extracted TL examples. The performance of our engine was evaluated in terms of *word error rate* (WER) with regard to a single oracle translation. The word error rate is computed as a Levenstein distance (counting the same penalty for *insertion*, *deletion* and *substitution* operations).

With our current decoder implementation, decoding over the full search space becomes impractical as soon as the sentence to translate contains more than 10 words. Therefore we resorted to several pruning strategies (the description of which is irrelevant in this context), yielding a configuration that translates reasonably fast enough without too many detrimental effects on the quality.

We ran seven translation sessions corresponding to different ways of selecting among the TL proposals. The results of these translation sessions are summarized in Table 2. In this table, MERGE- F_n corresponds to a translation session where the n most frequent TL proposals returned by a given query are considered; MERGE- S_n corresponds to a session where the n best-ranked alignments (scored by equation 1) are considered; SMT corresponds to a session in which the statistical engine operated alone without extracted examples. In this experiment, we tested three values of n : 3, 5, and 10.

system	WER	system	WER	system	WER
SMT	68.9%				
MERGE-F3	73.9%	MERGE-F5	74.2%	MERGE-F10	74.2%
MERGE-S3	75.4%	MERGE-S5	74.9%	MERGE-S10	74.4%

Table 2. Translation performance of the SMT engine alone (line 1) and TM examples under different scenarios.

Much to our disappointment, all the attempts to merge the extracted examples into the decoder resulted in an increase of the overall WER (around 5%).

It is not easy to evaluate whether this drop in performance also reflects a significant loss in the quality of the translation. Figure 5 gives two examples where we observed an improvement in WER after merging³

These examples call for some comment. The first one illustrates the situation where some words are not known to the statistical engine (here the person name *raymonde folco*), but present in the translation memory. Clearly, this is a situation where our approach should yield noticeable improvements.

The second example may help explain the measured degradations. First, there are some examples that are irrelevant to the translation (*e.g. but no authority/le front des soins*), obviously a bad alignment. Second, some examples are only partially good (*e.g. all the responsibilities / détient toutes les responsabilités*), however, our merging strategy only considered the complete TL examples. Last but not least, there are some SL sequences that we may not want to consider, as for example the sequence *they have* for which we only obtained vague translations. A simple filter could reject such undesired queries and hopefully improve the results.

6 Discussion

Although the results presented in the above evaluation are somewhat disappointing, we feel that there are positive aspects to our experiments. The output of the translation sessions shows many cases where the translation obtained by merging the extracted examples with the decoder clearly improved the results obtained by the engine alone. One possible explanation is that an evaluation based on the WER metric and single oracle translations might not fully do justice to the real contribution of the TM.

Yet, it is legitimate to ask whether the approach presented here does not involve a vicious circle, since both the extraction of TL examples from the TM and the translation engine rely on similar types of statistical translation models, essentially trained and used on the same material. In this regard, it is interesting to note that in the TM matching phase, the statistical models are used to perform “translation analysis”, while the decoder does “translation generation”. As they currently stand, statistical language and translation models are very crude devices. One of the assumptions underlying this work is that given their inherent weaknesses, the former task (analysis) is easier in practice than the latter (generation).

As Daniel Marcu points out [9], improving SMT outputs with TM examples is only possible insofar as it compensates for the imperfections of existing models and decoders. It supposes that the TM contains good translations for SL sequences, which the decoder would not normally produce, either because of the reduced search-space within which it operates, or because of these translations’ low frequency in the training corpus. But whether or not the decoder is able to take advantage of the better translations contained in the TM crucially depends on several aspects.

³ The full translation sessions are available at

<http://www.iro.umontreal.ca/~felipe/ResearchOutput/AMTA2002>.

SRC	ms. raymonde folco (parliamentary secretary to the minister of human resources development , lib .)
REF	mme raymonde folco (secrétaire parlementaire de la ministre du développement des ressources humaines , lib .)
MERGE-F3	mme raymonde folco (secrétaire parlementaire de la ministre du développement des ressources humaines .) , [wer=15.7%]
SMT	mme UNKNOWN clark (secrétaire parlementaire du ministre des finances et des ressources humaines .) [wer=47.4%]

EXAMPLES	ms. raymonde folco (parliamentary secretary to the minister of human resources development/ mme raymonde folco (secrétaire parlementaire de la ministre du développement des ressources humaines
----------	--

SRC	they have all the responsibilities but no authority .
REF	ils ont toutes les responsabilités , mais aucune autorité .
MERGE-F3	ils se faire le tour des responsabilités , mais peu de pouvoirs . [wer=61.5%]
SMT	ils ont tous la responsabilité d' emprunt non . [wer=72.7%]

EXAMPLES	
	<i>all the responsibilities</i> / détient toutes les responsabilités
	<i>all the responsibilities</i> / faire le tour des responsabilités
	<i>all the responsibilities</i> / les communications
	<i>they have</i> / ils se
	<i>they have</i> / ils ont réussi
	<i>they have</i> / ils ont passé
	<i>but no authority</i> / le front des soins
	<i>but no authority</i> / , mais peu de pouvoirs

Fig. 5. Translation outputs and matching examples. SRC designates the SL sentence; REF indicates the oracle translation and EXAMPLES indicates the examples given to the decoder.

First, looking up SL sequences *verbatim* is admittedly a rather simplistic scheme – one we essentially viewed as a starting point to gauge the potential of the approach. Macklovitch and Russell [7] convincingly argue in favor of performing more “linguistically informed” searches, for instance taking inflectional morphology and syntax into account, or dealing with named entities and numerical expressions in a sensible way. In a more general way, much work in EBMT could be of use in a setup such as ours (for example, see [4]).

Also, a close inspection of TL examples reveals that incorrect alignments are often to blame for bad translations. In particular, such imprecise alignments as those in Figure 5 exacerbate the *boundary friction* problem, well-known in EBMT circles. We are currently experimenting with more elaborate alignment techniques.

Both our strategies of selecting TM examples on the basis of their frequency or alignment score typically prevent the decoder from picking low-frequency examples, in favor of more literal ones. Alternative ranking techniques are needed to prevent this kind of systematic behavior. For one thing, we do not currently force the bracketing of SL sequences found in the TM to match that of the

input sequence. This would probably help filtering out irrelevant or “syntactically incompatible” matches. In the same vein, we could favor TM couples that either globally resemble the input sentence, or that have “syntactic similarities” around the boundaries of the matching sequence.

Finally, and as mentioned earlier, there are many more ways of feeding the TM examples to the translation engine. In short, the time to throw in the towel has not yet come.

Acknowledgments

We would like to thank Elliott Macklovitch and George Foster for the fruitful comments they made on this work. The statistical models used in this work were built using software written by George Foster.

References

1. Anne Abeillé, Lionel Clément and Alexandra Kinyon. Building a treebank for French. *International Conference on Language Resources & Evaluation (LREC)*, Athens, Greece (2000). 109
2. Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer. The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*, 19-2 (1993) 263–311. 107, 107
3. Ralf D. Brown. Example-Based Machine Translation in the Pangloss System. *International Conference on Computational Linguistics (COLING)*, Copenhagen, Denmark, (1996) 169–174. 105
4. Michael Carl and Silvia Hansen. Linking Translation Memories with Example-Based Machine Translation. *Machine Translation Summit VII*, Singapore (1999) 617–624. 112
5. George F. Foster. *Statistical Lexical Disambiguation*. MSc thesis, McGill University, School of Computer Science (1991). 108
6. Philippe Langlais. Opening Statistical Translation Engines to Terminological Resources. *7th International Workshop on Applications of Natural Language to Information Systems (NLDB)*. June 27-28, 2002. Stockholm, Sweden (2002). 104, 104
7. Elliot Macklovitch and Graham Russell. What’s been Forgotten in Translation Memory. *The Association for Machine Translation in the Americas (AMTA-2000)*, Cuernavaca, Mexico (2000). 112
8. Elliott Macklovitch, Michel Simard and Philippe Langlais. TransSearch: A Free Translation Memory on the World Wide Web. *International Conference on Language Resources & Evaluation (LREC)*, Athens, Greece (2000) 641–648. 105
9. Daniel Marcu. Towards a Unified Approach to Memory- and Statistical-Based Machine Translation. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France (2001) 378–385. 111
10. S. Niessen, S. Vogel, H. Ney and C. Tillmann. A DP based Search Algorithm for Statistical Machine Translation. *COLING/ACL* (1998) 960–966. 107, 107
11. Miles Osborne. Shallow Parsing as Part-of-Speech Tagging. *Proceedings of CoNLL*, Lisbon, Portugal (2002). 105, 108
12. Michel Simard, George Foster and Pierre Isabelle. Using Cognates to Align Sentences in Bilingual Corpora. *Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Montréal, Québec (1992) 67–82. 108
13. Michel Simard and Philippe Langlais. Sub-sentential Exploitation of Translation Memories. *MTSummit-VIII*, Santiago de Compostela, Spain, (2001) 335–340. 104, 105, 109, 109

Classification Approach to Word Selection in Machine Translation

Hyo-Kyung Lee

Department of Computer Science, 1304 W. Springfield Ave.,
Urbana, IL 61801, USA
`h-lee5@cs.uiuc.edu`

Abstract. We present a classification approach to building a English-Korean machine translation (MT) system. We attempt to build a word-based MT system from scratch using a set of parallel documents, on-line dictionary queries, and monolingual documents on the web. In our approach, MT problem is decomposed into two sub-problems — word selection problem and word ordering problem of the selected words. In this paper, we will focus on the word selection problem and discuss some preliminary results.

1 Introduction

There are billions of on-line documents available on the web nowadays. As the volume of such documents increases, so does the importance of machine translation since languages used in them vary. Thanks to the abundance in on-line documents available, machine learning techniques are popular in several natural language processing tasks and MT is no exception. It is often believed that a MT system requires a huge amount of very sophisticated knowledge about the correspondences between two words, phrases, sentences, and idioms in the source and target languages. Its performance is limited to a great extent by the ability to encode such information either manually or automatically. Our suggestion is to encode such knowledge implicitly and automatically using a large network of classifiers via learning.

Here we concentrate on the word selection task in MT. The real world MT systems often sacrifice the processing of contexts to provide quick translations but our approach is both robust and efficient for the high dimensionality requirement of the context handling. This fact makes our system very appealing for the success of statistical MT in real use.

2 Statistical Machine Translation and Word Selection Problem

In this section, we present our overall framework for the statistical machine translation problem first and then the role of the word selection problem by providing some examples from real data.

2.1 Statistical Machine Translation Problems

Given a set of two equivalent documents $C_{(s,t)}^P = \{(D_s, D_t)_\infty\}$ and two monolingual corpora $C_s^M = \{(D_s)_\infty\}$ and $C_t^M = \{(D_t)_\infty\}$ in a source language L_s and a target language L_t , the statistical MT problem can be modeled as learning a set of classifiers $\mathcal{F}$ that can map words in a source language $\{s_1, s_2, \dots, s_J\} \in L_s$ to words in a target language $\{t_1, t_2, \dots, t_I\} \in L_t$: $\mathcal{F}_i(\{s_j\}) \rightarrow t_i$. The complexity of translation depends on $|L_s|, |L_t|, |\mathcal{F}|$ and the curse of high dimensionality prevails in any realistic MT system. Furthermore, the difficulty of learning in MT originates from the following two ubiquitous problems.

First, the word selection problem:

$$s_j \rightarrow t_1 \vee t_2 \dots \vee t_n$$

in which a word s_j can have several different meanings depending on its context $\{s_{-\infty}, \dots, s_{j-1}, s_{j+1}, \dots, s_\infty\}$. This problem becomes more complicated if we consider the fertility problem, in which translations are often performed at the phrase level and some words appear or disappear automatically during translation [10].

Second, the word order problem:

$$s_i s_j \rightarrow t_k t_l$$

where $i, j, k, l (i < j, k < l)$ stands for the position in the sentence given translations $s_i \rightarrow t_l, s_j \rightarrow t_k$. One good example is the word order change during the translation of English into Korean. English has a subject-verb-object word order but Korean has a subject-object-verb word order. This work will focus on the first problem only and we'll explain it further in the next subsection.

2.2 Word Selection Problem

Solving the word selection problem is a very important step in performing high quality machine translation, and it has implications as a stand-alone problem. Yet, it is also very difficult to solve because its sensitivity to the local syntax and semantics. To illustrate the importance of the word selection problem clearly, we pick several English sentences that have the word *scores* in different contexts and provide its corresponding actual Korean translation in Table 1. If such an ambiguous word is not resolved properly, depending on its context, the result of translation becomes awkward and meaningless.

There is another interesting source of ambiguity in real translation that is language specific. In Table 2 there are four different translation forms of the word *age* depending on the object to which the word is related. Although the Korean words *Yun-Soo* and *Yun-Ryung* are the same in their meaning, they are used in different contexts. The word *Yun-Soo* is appropriate only for an inanimate object like *tire* and *Yun-Ryung* for humans like *children*. In this respect, the word selection problem is somewhat different from the word sense disambiguation task.

Table 1. Korean translations of the word **scores** — WSD-like word selection.

Sentence in Korea Herald news article	Translation
Recently both your editorial writer and a female contributor to <i>In My View</i> expressed utter dismay and disbelief that Korean veterans have gotten so angry over the constitutional court’s decision to strike down a policy to award bonus points to the test scores of former soldiers who apply for low-level government jobs.	<i>Sung-Juk</i>
Every year scores of young Korean men are killed while performing tasks such as serving in flood rescue operations and fighting off rabid demonstrators on the streets of Seoul.	<i>Su-Ship-Myung</i>
“They have a huge library for notes and scores , and can get money from the Net,” said WestLB’s Tokyo-based Internet analyst Ortwin Gierhake.	<i>Eum-Ak</i>

Table 2. Korean translations of the word **age** — Non WSD-like contextual differences.

Sentence in Korea Herald news article	Translation
John Rintamaki, Ford’s group vice president and chief of staff, said the tires would be replaced according to their age with, Ford to contact affected customers by mail.	<i>Yun-Soo</i>
I think my children’s table manners are appropriate for their age , and they are now two years older than the last time she saw them.	<i>Yun-Ryung</i>
To remedy the situation, parents spend more than 7 trillion <i>Won</i> a year on private lessons, and an increasing number of them choose to send their children to schools in advanced nations at an early age or emigrate with them for the sake of their education.	<i>Cho-Gi</i>
When “Jeremy” first asked me out, I said, “Are you serious? I have children your age .” He said, “So what?” and kept calling.	<i>Na-Yi</i>

3 Classification Approach to Word Selection Problem

In this section, we recast the word selection problem in MT as a learning problem over a feature space to resolve ambiguities among possible translations as suggested in [15]. We suggest to view the problem of word selection as that of choosing, given s and its context c_s , one of the dictionary entries $\{t_1, t_2, \dots, t_k\}$ corresponding to s . The context will be transformed into a feature space and a classifier f_s will be trained to make the selection for each word of interest s .

3.1 Translation Dictionary

A translation dictionary $\mathcal{D}$, which is a special instance of $C_{(s,t)}^P$ in section 2, is assumed to have all k possible translations for each word s_j in a source document D_s . Thus, $\mathcal{D}$ is defined as

$$\mathcal{D} = \bigcup_j \{(\{s_j\}, \{t_{j1}, t_{j2}, \dots, t_{jk}\})\}.$$

This dictionary is exactly same as the concept of confusion set in the context-sensitive spelling correction problem [5] but the size of each confusion set, k , is not binary and can vary depending on the fertility of the source word s_j . For example, we can easily construct a translation dictionary $\{(\{\text{scores}\}, \{\textit{Sung-Juk}, \textit{Su-Ship-Myung}, \textit{Eum-Ak}\}), (\{\text{age}\}, \{\textit{Yun-Soo}, \textit{Yun-Ryung}, \textit{Cho-Gi}, \textit{Na-Yi}\})\}$ from Table 1 and Table 2. Once the translation dictionary is constructed, the word selection problem becomes a simple multi-class classification problem where each t_{jk} plays the role of class label.

3.2 Portable Features for Classification

In general, the use of expressive and large sets of features leads to a good classification performance especially when the features in such sets are directly relevant to the classification task. We use a feature extraction mechanism similar to that described in context-sensitive spelling correction [5]. However, the POS tags information is not considered in the feature set due to the portability issues of our technique [7]. That is, we are interested in observing how well it can select the right translation from the context words only since not all languages have POS taggers and share the same POS tagging scheme. To maximize the portability, all features used in this work are sparse collocations of two words which means all possible, not limited to the next word, bigrams within the sentence as shown in Table 3.

3.3 SNoW Classifier

The SNoW (Sparse Network of Winnows) learning architecture is a multi-class classifier that is suitable for large scale learning tasks [2]. We want to use it to solve the word selection problem because the central learning algorithm of

Table 3. Features used in the word selection problem. Example features are extracted from a source sentence *I am a boy*.

Features	Description	Example
Word	Each word in a source sentence	I,am,a,boy
Collocation	Two consecutive words	I-am, am-a, a-boy
Sparse Collocation	Any two words that appear together	I-a, I-boy, am-boy

SNoW architecture, Winnow, is a feature-efficient and mistake-bound learning algorithm [8]. This classifier has been successfully applied to several other natural language processing problems such as shallow parsing [11], part of speech (POS) tagging [14] and spelling correction tasks [5]. We are experimenting with the classifier for the MT task for the first time.

The SNoW classifier is a network of linear threshold functions with multiplicative update rules. A linear threshold function in SNoW is a weighted sum over pre-defined features and it becomes active if and only if

$$\sum_{f \in \mathcal{F}} w_f > \theta$$

where each feature f in an active feature set $\mathcal{F}$ is extracted from an example (i.e. source sentence) through a feature extractor. The word *Sparse* in SNoW means that each class label (i.e. each target word t_{jk}) is learned as a function of a subset of all the features known to the classifier.

The weights are updated in both mistake-driven and on-line manner. Whenever the selection of the translation word is wrong, the weights of active linear functions are either promoted or demoted by multiplication parameter $\alpha (> 1)$. This multiplicative update rule, called Winnow, is an ideal choice for tasks that involve a high dimensionality requirement because the number of examples it requires to learn a linear function grows linearly with the number of relevant features and only logarithmically with the total number of features.

4 Experiments

In this section, we present the real application of our approach and show some preliminary results.

4.1 Data

A total of 689 translated documents dated between November 2000 and January 2002 were obtained from the www.koreaherald.com news web site. The entire archive consists of 17846 sentences and 316638 words counting English and Korean together. We performed experiments on the most confused 121 nouns for the word selection problem. It is the same set used in the word sense disambiguation task described in [12].

In statistical MT, all possible translations of a word can be obtained only by looking at real translation data, aligned at the word level. Plain on-line dictionaries available on the Internet are helpful but they do not have the complete set of all possible translations [6]. In this paper, we will not discuss the alignment problem, because all data is manually aligned with a little help of on-line dictionaries, and all possible translations are obtained directly from the translated documents.

4.2 Results

The results are shown in Table 4, where k means the size of the set that contains all possible translations of each noun and m means the total number of example sentences in the corpus. Among the 121 nouns experimented with, we are showing only the result of the nouns that have more than 50 example sentences in the corpus. We also pruned sentences such that the target translation appears less than three times in the corpus.

We first trained and tested the classifier on the same data set because there is not enough data to separate the dataset in a standard 80%-20% fashion. We provided the result of baseline classifier (i.e. selecting the most frequent translation) and naïve Bayes classifier for comparison. Finally, we performed 1-fold cross-validation to overcome the data sparseness problem and to see how well our classifiers work.

4.3 Word Sense Disambiguation

To support the strength of our approach, we also performed word sense disambiguation(WSD) experiments on English and Korean SENSEVAL-2 data¹. We tested our classifier on the English lexical sample and Korean lexical sample tasks with the same feature set defined in Table 3. The result indicates that our method is close to the baseline and our *word-only* approach might not be suitable for WSD tasks. As Pederson pointed out [13], the poor result in English sample task should be attributed to the use of a less expressive feature set rather than the learning algorithm itself. It would be interesting if we could test our classifier with the same syntactic (i.e. POS tags) or semantic (i.e. WordNet) features that are incorporated in the best SENSEVAL-2 system.

We've also found that the SENSEVAL-2 data share some words that we used for word selection. Table 6 demonstrates that there seems to be no direct correlation between word senses and translation word selection. The discrepancy between k in Table 4 and the number of translations in Table 6 is due to the use of pruned data set in the experiment.

5 Discussion

Building a statistical MT system with similarly scarce resources is described in several recent projects and is getting more attention among MT researchers [9,4,6,1].

¹ <http://www.sle.sharp.co.uk/senseval2/>

Table 4. Word Selection in English-Korean Machine Translation. k means the number of possible translations and m is the number of examples in the corpus. Testing was performed with 1-fold cross validation method.

Noun	k	m	Training			Testing	
			Baseline	NaïveBayes	SNoW	NaïveBayes	SNoW
case	4	57	42.11	28.07	92.98	31.57	43.86
city	3	54	57.41	12.96	94.44	12.96	59.23
company	5	54	47.27	22.22	87.04	33.33	48.15
country	11	141	40.43	7.092	73.76	10.64	41.84
day	5	135	36.30	40.74	97.04	41.48	45.73
end	6	65	38.46	41.54	95.38	40.00	44.61
family	4	89	58.43	64.04	77.53	57.30	58.43
history	2	47	93.62	93.62	93.62	93.62	93.62
home	10	87	41.38	9.195	88.51	11.49	43.67
house	4	53	64.15	66.04	77.36	66.04	66.04
information	2	61	93.44	93.44	93.44	93.44	93.44
law	2	75	96.00	96.00	96.00	96.00	96.00
life	7	62	30.65	56.45	90.32	30.64	25.81
man	7	58	39.66	25.86	94.83	29.31	36.21
money	4	81	77.78	82.72	90.12	69.01	76.54
month	3	93	83.87	83.87	91.40	85.91	83.87
nation	6	114	30.70	27.19	99.12	22.81	28.94
number	7	72	40.28	37.50	91.67	23.61	28.57
part	6	59	42.37	49.15	91.53	38.98	44.07
party	3	80	88.75	91.25	90.00	88.23	87.50
plan	3	51	88.24	94.12	92.20	88.88	88.23
policy	2	98	96.94	96.94	96.94	96.94	96.94
power	7	52	30.77	28.85	94.23	29.16	46.15
problem	2	74	93.24	97.30	93.24	94.00	100.0
public	8	97	44.33	26.80	84.54	35.48	47.42
school	4	45	75.56	82.22	84.44	75.55	77.78
state	9	93	33.33	9.677	94.62	24.72	55.91
system	6	58	25.86	24.14	91.38	22.22	35.08
time	13	237	22.78	23.63	97.89	23.21	40.08
way	7	95	36.84	20.00	94.74	22.82	36.56
world	4	120	75.83	80.83	85.83	75.83	75.73
Weighted Average			53.87	48.61	91.05	47.49	57.46

Table 5. SENSEVAL-2 Experiment with SNoW. English baseline is the *Commonest* baseline.

Language	Baseline (Precision/Recall)	Best (Precision/Recall)	SNoW (Precision/Recall)
English	0.476/0.476	0.642/0.642	0.471/0.471
Korean	0.685/0.714	0.698/0.74	0.687/0.73

Table 6. Sense vs. Translation

Word	No. of Senses	No. of Translations
art	7	3
child	6	9
church	4	2
day	7	12
nation	4	9
sense	7	15

Basically, we are attempting to extend the linear classifier learning approach described in the context sensitive spelling correction problem [5] and the sequential model for multi-class classification [3] into larger scale. It is well understood today that many of the classifiers that are used in natural language processing tasks are linear classifier [15]. The naïve Bayes algorithm also classifies using a linear hypothesis [16] that has somewhat limited expressivity, given that the weights are simple statistics over the instance space. Therefore, it is unsurprising and well-documented that discriminative algorithms such as the Winnow variation in SNoW perform significantly better than naïve Bayes.

The results clearly demonstrate that there are certain limits in the knowledge-poor word-only based classification method, which is already confirmed in the previous work by Koehn and Knight [6]. Although the straightforward comparison to their work is difficult due to the small amount of data that we have, they also could achieve only 3.7% improvement over the most frequent method with monolingual corpus and lexicon by using EM algorithm. Such a method can be easily extended to any human language but it needs a huge amount of examples to be useful in PAC-learnability sense [17]. Therefore, there is a trade-off between the amount of examples needed and the portability of feature generation mechanism. Together with the result of WSD experiment, the best performance can only be achieved by the use of good classifier like SNoW and the use of right morphological, syntactic, and semantic features.

Definitely, we are interested in extending this classification framework into the realm of the word ordering problem and word-level alignment problems. In the ordering problem, the inference with the classifiers is a major challenge because classifiers need to interact with each other during the ordering process. In the word-level alignment problem, generating almost 100% accurate alignment is critical to get a complete translation dictionary $\mathcal{D}$. Otherwise, our approach cannot scale up well automatically for real-world large lexicons, since the small errors in word-level alignment will magnify the final translation errors in the end.

6 Conclusion

We have presented a learning approach to word selection in English-Korean machine translation. The SNoW learning architecture is an ideal choice for the word selection problem in MT, which requires context sensitivity because the task demands a capability of processing high dimensionality in the feature space. Since we cannot explicitly specify the word selection rules for every word, it is better to use a learning approach and store the rules in a network of classifiers implicitly. It is our observation that the word selection problem in MT is not exactly the same as the word sense disambiguation task for two reasons. First, although a source word can have multiple senses, it can have only one translation that also has multiple senses in the target language. Second, a word can have multiple forms in translation for the same sense depending on its context. The above observation leads us to conclude that the creation of translation dictionaries directly from a corpus is a must, and an efficient context-sensitive classifier is needed for a high quality translation job. Therefore, our approach is quite promising for the large-scale real-world application of MT, because it is based on a well-known, attribute-efficient, and robust learning algorithm. Yet, generating the right amount of *universally effective* features still remains as an open problem.

7 Acknowledgments

The author is grateful to Dan Roth, Chad Cumby and anonymous referees for their useful discussions and comments. This work was partially supported by NSF grants CAREER IIS-9984168 and ITR IIS-0085836 and ONR MURI Award.

References

1. Bangalore, S., Ricciardi, G.: A finite-state approach to machine translation. In: NAACL. (2001) 119
2. Carlson, A., Cumby, C., Rosen, J., Roth, D.: The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department (1999) 117
3. Even-Zohar, Y., Roth, D.: A sequential model for multi-class classification. In: EMNLP. (2001) 121
4. Germann, U.: Building a statistical machine translation system from scratch: How much bang can we expect for the buck. In: Proceedings of the Data-Driven MT Workshop of ACL-01. (2001) 119
5. Golding, A., Roth, D.: A winnow-based approach to spelling correction. Machine Learning 34 (1999) 107–130 117, 117, 118, 121
6. Koehn, P., Knight, K.: Knowledge sources for word-level translation models. In: Empirical Methods in Natural Language Processing conference. (2001) 119, 119, 121
7. Lee, H.: A theory of portability. In: LREC2002: Workshop on Portability Issues in HLT. (2002) 117

8. Littlestone, N.: Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning* **2** (1988) 285–318 118
9. Marcu, Y.A.O.U.G.U.H.K.K.P.K.D., Yamada, K.: Translating with scarce resources. In: *National Conference on Artificial Intelligence (AAAI)*. (2000) 119
10. Marcu, U.G.M.J.K.K.D., Yamada, K.: Fast decoding and optimal decoding for machine translation. In: *Proc. of the Conference of the Association for Computational Linguistics (ACL)*. (2001) 115
11. Munoz, M., Punyakanok, V., Roth, D., Zimak, D.: A learning approach to shallow parsing. In: *EMNLP-VLC'99, the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. (1999) 168–178 118
12. Ng, H.T., Lee, H.B.: Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In: *Proc. of 34th Conference of the ACL*. (1996) 118
13. Pederson, T.: Evaluating the effectiveness of ensembles of decision trees in disambiguating senseval lexical samples. In: *ACL02: Workshop on WSD: Recent Successes and Future Directions*. (2002) 119
14. Roth, D., Zelenko, D.: Part of speech tagging using a network of linear separators. In: *COLING-ACL 98, The 17th International Conference on Computational Linguistics*. (1998) 1136–1142 118
15. Roth, D.: Learning to resolve natural language ambiguities: A unified approach. In: *Proc. of the American Association of Artificial Intelligence*. (1998) 806–813 117, 121
16. Roth, D.: Learning in natural language. In: *Proc. of the International Joint Conference on Artificial Intelligence*. (1999) 898–904 121
17. Valiant, L.G.: A theory of the learnable. *Communications of the ACM* **27** (1984) 1134–1142 121

Better Contextual Translation Using Machine Learning

Arul Menezes

Microsoft Research, One Microsoft Way, Redmond WA 98008, USA
arulm@microsoft.com

Abstract: One of the problems facing translation systems that automatically extract transfer mappings (rules or examples) from bilingual corpora is the trade-off between contextual specificity and general applicability of the mappings, which typically results in conflicting mappings without distinguishing context. We present a machine-learning approach to choosing between such mappings, using classifiers that, in effect, selectively expand the context for these mappings using features available in a linguistic representation of the source language input. We show that using these classifiers in our machine translation system significantly improves the quality of the translated output. Additionally, the set of distinguishing features selected by the classifiers provides insight into the relative importance of the various linguistic features in choosing the correct contextual translation.

1 Introduction

Much recent research in machine translation has explored data-driven approaches that automatically acquire translation knowledge from aligned or unaligned bilingual corpora. One thread of this research focuses on extracting transfer mappings, rules or examples, from parsed sentence-aligned bilingual corpora [1,2,3,4]. Recently this approach has been shown to produce translations of quality comparable to commercial translation systems [5].

These systems typically obtain a dependency/predicate argument structure (called “logical form” in our system) for source and target sentences in a sentence-aligned bilingual corpus. The structures are then aligned at the sub-sentence level. From the resulting alignment, lexical and structural translation correspondences are extracted, which are then represented as a set of transfer mappings, rules or examples, for translation. Mappings may be fully specified or contain “wild cards” or under-specified nodes.

A problem shared by all such systems is choosing the appropriate level of generalization for the mappings. Larger, fully specified mappings provide the best contextual¹ translation, but can result in extreme data sparsity, while smaller and more under-specified mappings are more general, but often do not make the necessary contextual translation distinctions.

¹ In this paper, context refers only to that within the same sentence. We do not address issues of discourse or document-level context.

All such systems (including our own) must therefore make an implicit or explicit compromise between generality and specificity. For example, Lavoie [4] uses a hand-coded set of language-pair specific alignment constraints and attribute constraints that act as templates for the actual induced transfer rules.

As a result, such a system necessarily produces many mappings that are in conflict with each other and do not include the necessary distinguishing context. A method is needed, therefore, to automatically choose between such mappings.

In this paper, we present a machine-learning approach to choosing between conflicting transfer mappings. For each set of conflicting mappings, we build a decision tree classifier that learns to choose the most appropriate mapping, based on the linguistic features present in the source language logical form. The decision tree, by selecting distinguishing context, in effect, selectively expands the context of each such mapping.

2 Previous Work

Meyers [6] ranks acquired rules by frequency in the training corpus. When choosing between conflicting rules, the most frequent rule is selected. However, this will result in choosing the incorrect translation for input for which the correct contextual translation is not the most frequent translation in the corpus.

Lavoie [4] ranks induced rules by log-likelihood ratio and uses error-driven filtering to accept only those rules that reduce the error rate on the training corpus. In the case of conflicting rules this effectively picks the most frequent rule.

Watanabe [7] addressed a subset of this problem by identifying “exceptional” examples, such as idiomatic or irregular translations, and using such examples only under stricter matching conditions than more general examples. However he also did not attempt to identify what context best selected between these examples.

Kaji [1] has a particularly cogent exposition of the problem of conflicting translation templates and includes a template refinement step. The translation examples from which conflicting templates were derived are examined, and the templates are expanded by the addition of extra distinguishing features such as semantic categories. For example, he refines two conflicting templates for “play <NP>” which translate into different Japanese verbs, by recognizing that one template is used in the training data with sports (“play baseball”, “play tennis”) while the other is used with musical instruments (“play the piano”, “play the violin”). The conflicting templates are then expanded to include semantic categories of “sport” and “instrument” on their respective NPs. This approach is likely to produce much better contextual translations than the alternatives cited. However, it appears that in Kaji’s approach this is a *manual* step, and hence impractical in a large-scale system.

The approach described in this paper is analogous to Kaji, but uses machine learning instead of hand-inspection.

3 System Overview

3.1 The Logical Form

Our machine translation system [5] uses logical form representations (LFs) in transfer. These representations are graphs, representing the predicate argument structure of a sentence. The nodes in the graph are identified by the lemma (base form) of a content word. The edges are directed, labeled arcs, indicating the logical relations between nodes.

Additionally, nodes are labeled with a wealth of morpho-syntactic and semantic features extracted by the source language analysis module.

Logical forms are intended to be as structurally language-neutral as possible. In particular, logical forms from different languages use the same relation types and provide similar analyses for similar constructions. The logical form abstracts away from such language-particular aspects of a sentence as voice, constituent order and inflectional morphology. Figure 1 depicts an example Spanish logical form, including features such as number, gender, definiteness, etc.

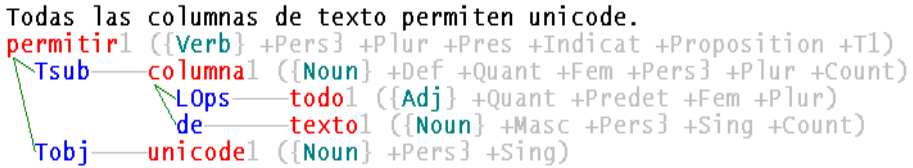


Figure 1: Example Logical Form

3.2 Acquiring and Using Transfer Mappings

In our MT architecture, alignments between logical form subgraphs of source and target language are identified in a training phase using aligned bilingual corpora. From these alignments a set of transfer mappings is acquired and stored in a database. A set of language-neutral heuristic rules determines the specificity of the acquired mappings. This process is discussed in detail in [8].

During the translation process, a sentence in the source language is analyzed, and its logical form is matched against the database of transfer mappings. From the matching transfer mappings, a target logical form is constructed which serves as input to a generation component that produces a target string.

3.3 Competing Transfer Mappings

It is usually the case that multiple transfer mappings are found to match each input sentence, each matching some subgraph of the input logical form. The subgraphs matched by these transfer mappings may overlap, partially or wholly.

Overlapping mappings that indicate an identical translation for the nodes and relations in the overlapping portion are considered *compatible*. The translation system can merge such mappings when constructing a target logical form.

Overlapping transfer mappings that indicate a different translation for the overlapping portions are considered *competing*. These mappings cannot be merged when constructing a target logical form, and hence the translation system must choose between them. Figure 2 shows an example of two partially overlapping mappings that compete on the word “presentar”. The first mapping translates “presentar ventaja” as “have advantage”, whereas the second mapping translates “presentar” by itself to “display”.

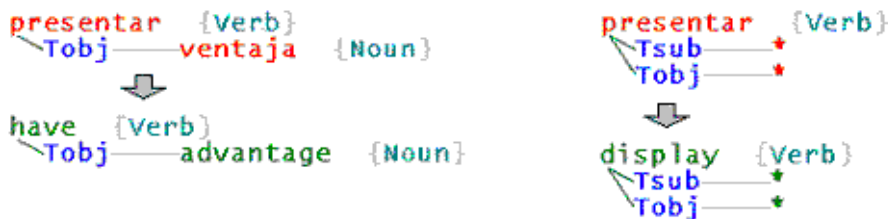


Figure 2: Competing transfer mappings

3.4 Conflicting Transfer Mappings

In this paper we examine the subset of competing mappings that overlap fully. Following Kaji [1] we define *conflicting mappings* as those whose left-hand sides are identical, but whose right-hand sides differ.

Figure 3 shows two conflicting mappings that translate the same left-hand side, “todo (las) columna(s)”, as “all (of the) column(s)” and “(the) entire column” respectively.

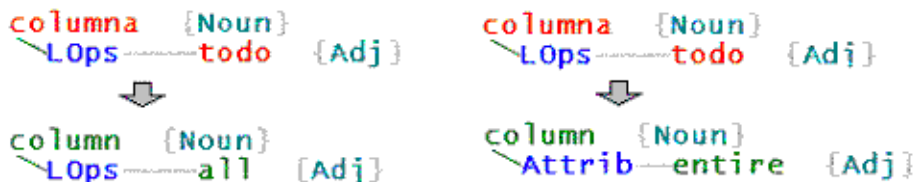


Figure 3: Conflicting transfer mappings

4 Data

Our training corpus consists of 351,026 aligned Spanish-English aligned sentence pairs taken from published computer software manuals and online help documents. The sentences have an average length of 17.44 words in Spanish and 15.10 words in English. Our parser produces a parse in every case, but in each language roughly 15% of the parses produced are “fitted” or non-spanning. We apply a conservative heuristic

and only use in alignment those sentence-pairs that produced spanning parses in both languages. In this corpus, 277,109 sentence pairs (or 78.9% of the original corpus) were used in training.

5 Using Machine Learning

The machine learning approach that we use in this paper is a decision tree model. The reason for this choice is a purely pragmatic one: decision trees are easy to construct and easy to inspect. Nothing in our methodology, however, hinges on this particular choice².

We use a set of automated tools to construct decision trees [9] based on the features extracted from logical forms.

5.1 The Classification Task

Each set of conflicting transfer mappings (those with identical left-hand sides) comprises a distinct classification task. The goal of each classifier is to pick the correct transfer mapping. For each task the data consists of the set of sentence pairs where the common left-hand side of these mappings matches a portion of the source (Spanish) logical form of the sentence pair. For a given training sentence pair for a particular task, the correct mapping is determined by matching the (differing) right-hand sides of the transfer mappings with a portion of the reference target (English) logical form.

5.2 Features

The logical form provides over 200 linguistic features, including semantic relationships such as subject, object, location, manner etc., and features such as person, number, gender, tense, definiteness, voice, aspect, finiteness etc. We use all these features in our classification task.

The features are extracted from the logical form of the source sentence as follows:

1. For every source sentence node that matches a node in the transfer mapping, we extract the lemma, part-of-speech and all other linguistic features available on that node.
2. For every source node that is a child or parent of a matching node, we extract the relationship between that node and its matching parent or child, in conjunction with the linguistic features on that node.
3. For every source node that is a grandparent of a matching node, we extract the chain of relationships between that node and its matching grandchild in conjunction with the linguistic features on that node.

² Our focus in this paper is understanding this key problem in data-driven MT, and the application of machine learning to it, and not the learner itself. Hence we use an off-the-shelf learner and do not compare different machine learning techniques etc.

These features are extracted automatically by traversing the source LF and the transfer mapping. The automated approach is advantageous, since any new features that are added to the system will automatically be made available to the learner. The learner in turn automatically discovers which features are predictive. Not all features are selected for all models by the decision tree learning tools.

5.3 Sparse Data Problem

Most of the data sets for which we wish to build classifiers are small (our median data set is 134 sentence pairs), making overfitting of our learned models likely. We therefore employ smoothing analogous to average-count smoothing as described by Chen and Goodman [10], which in turn is a variant of Jelinek and Mercer [11] smoothing.

For each classification task we split the available data into a training set (70%) and a parameter tuning set (30%). From the training set, we build decision tree classifiers at varying levels of granularity (by manipulating the prior probability of tree structures to favor simpler structures). If we were to pick the tree with the maximal accuracy for each data set independently, we would run the risk of over-fitting to the parameter tuning data. Instead, we pool all classifiers that have the same average number of cases per mapping (i.e. per target feature value). For each such pool, we then evaluate the pool as a whole at each level of decision tree granularity and pick the level of granularity that maximizes the accuracy of the pool as a whole. We then choose the same granularity level for all classifiers in the pool.

5.4 Building Decision Trees

From our corpus of 277,109 sentence pairs, we extracted 161,685 transfer mappings. Among these, there were 7027 groups of conflicting mappings, amounting to 19,582 conflicting transfer mappings in all, or about 2.79 mappings per conflicting group. The median size of the data sets was 134 sentence pairs (training and parameter tuning) per conflicting group.

We built decision trees for all groups that had at least 10 sentence pairs, resulting in a total of 6912 decision trees³. The number of features emitted per data set ranged from 15 to 1775, with an average of 653. (The number of features emitted for each data set depends on the size of the mappings, since each mapping node provides a distinct set of features. The number of features also depends on the diversity of linguistic features actually present in the logical forms of the data set).

In total there were 1775 distinct features over all the data sets. Of these, 1363 features were selected by at least one decision tree model. The average model had 35.49 splits in the tree, and used 18.18 distinct features.

³ We discard mappings with frequency less than 2. A conflicting mapping group (at least 2 mappings) would have, at minimum, 4 sentence pairs. The 115 groups for which we don't build decision trees are those with 4 to 9 sentence pairs each.

The average accuracy of the classifiers against the parameter tuning data set was 81.1% without smoothing, and 80.3% with smoothing. By comparison the average baseline (most frequent mapping within each group) was 70.8%⁴.

Table 1: Training data and decision trees

Size of corpus used (sentence pairs)	277,109
Total number of transfer mappings	161,685
Number of conflicting transfer mappings	19,582
Number of groups of conflicting mappings	7,027
Number of decision tree classifiers built	6,912
Median size of the data set used to train each classifier	134
Total number of features emitted over all classifiers	1,775
Total number of features selected by at least one classifier	1,363
Average number of features emitted per data set	653
Average number of features used per data set	18.18
Average number of decision tree splits	35.49
Average baseline accuracy	70.8%
Average decision tree accuracy without smoothing	81.1%
Average decision tree accuracy with smoothing	80.3%

6 Evaluating the Decision Trees

6.1 Evaluation Method

We evaluated the decision trees using a human evaluation that compared the output of our Spanish-English machine translation system using the decision trees (WithDT) to the output of the same system without the decision trees (NoDT), keeping all other aspects of the system constant.

Each system used the same set of learned transfer mappings. The system without decision trees picked between conflicting mappings by simply choosing the most frequent mapping, (i.e., the baseline) in each case.

We translated a test set of 2000 previously unseen sentences. Of these sentences, 1683 had at least one decision tree apply.

For 407 of these sentences at least one decision tree indicated a choice other than the default (highest frequency) choice, hence a different translation was produced between the two systems. Of these 407 different translations, 250 were randomly selected for evaluation.

Seven evaluators from an independent vendor agency were asked to rate the sentences. For each sentence the evaluators were presented with an English reference

⁴ All of the averages mentioned in this section are weighted in proportion to the size of the respective data sets.

(human) translation and the two machine translations⁵. The machine translations were presented in random order, so the evaluators could not know their provenance.

Assuming that the reference translation was a perfect translation, the evaluators were asked to pick the better machine translation or to pick neither if both were equally good or bad. Each sentence was then rated -1 , 0 or 1 , based on this choice, where -1 indicates that the translation from NoDT is preferred, 1 indicates that WithDT is preferred, and 0 indicates no preference. The scores were then averaged across all raters and all sentences.

6.2 Evaluation Results

The results of this evaluation are presented in Tables 2a and 2b. In Table 2a, note that a mean score of 1 would indicate a uniform preference (across all raters and all sentences) for WithDT, while a score of -1 would indicate a uniform preference for NoDT. The score of 0.33 ± 0.093 indicates a strong preference for WithDT.

Table 2b shows the number of translations preferred from each system, based on the average score across all raters for each translation. Note that WithDT was preferred more than twice as often as NoDT.

The evaluation thus shows that in cases where the decision trees played a role, the mapping chosen by the decision tree resulted in a significantly better translation.

Table 2a: Evaluation results: Mean Score

	Score	Significance	Sample size
WithDT vs. NoDT	0.330 ± 0.093	>0.99999	250

Table 2b: Evaluation results: Sentence preference

	Number of sentences WithDT rated better	Number of sentences NoDT rated better	Number of sentences neither rated better
WithDT vs. NoDT	167 (66.8%)	75 (30%)	8 (3.2%)

7 Comparisons

The baseline NoDT system uses the same (highest-frequency) strategy for choosing between conflicting mappings as is used by Meyers et al [6] and by Lavoie et al [4]. The results show that the use of machine learning significantly improves upon this strategy.

⁵ Since the evaluators are given a high-quality human reference translation, the original Spanish sentence is not essential for judging the MT quality, and is therefore omitted. This controls for differing levels of fluency in Spanish among the evaluators.

The human classification task proposed by Kaji [1] points us in the right direction, but is impractical for large-scale systems. The machine learning strategy we use is the first automated realization of this strategy.

8 Examining the Decision Trees

One of the advantages of using decision trees to build our classifiers is that decision trees lend themselves to inspection, potentially leading to interesting insights that can aid system development. In particular, as discussed in Sections 1 and 2, conflicting mappings are a consequence of a heuristic compromise between specificity and generality when the transfer mappings are acquired. Hence, examining the decision trees may help understand the nature of that compromise.

We found that of a total of 1775 features, 1363 (77%) were used by at least one decision tree, 556 (31%) of them at the top-level of the tree. The average model had 35.49 splits in the decision tree, and used 18.18 distinct features. Furthermore, the single most popular feature accounted for no more than 8.68% of all splits and 10.4% of top-level splits.

This diversity of features suggests that the current heuristics used during transfer mapping acquisition strike a good compromise between specificity and generality. This is complemented by the learner, which enlarges the context for our mappings in a highly selective, case-by-case manner, drawing upon the full range of linguistics features available in the logical form.

9 An Example

We used DnetViewer [12], a visualization tool for viewing decision trees and Bayesian networks, to explore the decision trees, looking for interesting insights into problem areas in our MT system. Figure 4 shows a simple decision tree displayed by this viewer. The text has been enlarged for readability and each leaf node has been annotated with the highest probability mapping (i.e., the mode of the predicted probability distribution shown as a bar chart) at that node.

The figure depicts the decision tree for the transfer mappings for (*—Attrib—agrupado), which translates, in this technical corpus, to either “grouped”, “clustered”, or “banded”. The top-level split is based on the input lemma that matches the * (wild-card) node. If this lemma is “indice” then the mapping for “clustered” is chosen.

If the lemma is not “indice”, the next split is based on whether the parent node is marked as indefinite, which leads to further splits, as shown in the figure, based again on the input lemma that matches the wild-card node.

Examples of sentence pairs used to build this decision tree:

Los datos y el índice agrupado residen siempre en el mismo grupo de archivos.
The data and the clustered index always reside in the same filegroup.

Se produce antes de mostrar el primer conjunto de registros en una página de datos agrupados.

Occurs before the first set of records is displayed on a banded data page.

En el caso de páginas de acceso a datos agrupadas, puede ordenar los registros incluidos en un grupo.

For grouped data access pages, you can sort the records within a group.

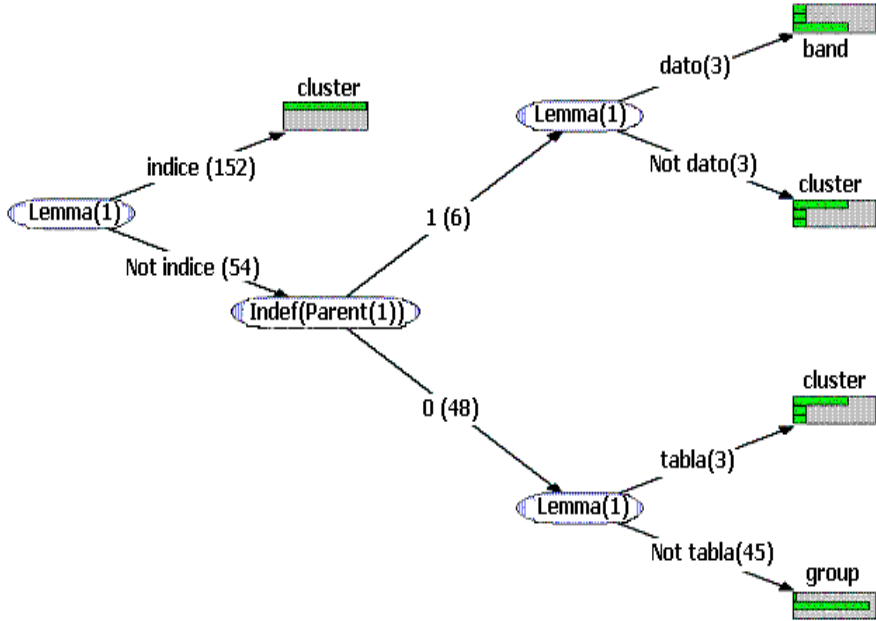


Figure 4: Decision tree for (*—Attrib—agrupado)

10 Conclusions and Future Work

We have shown that applying machine learning to this problem resulted in a significant improvement in translation over the highest-frequency strategy used by previous systems.

However, we built decision trees only for *conflicting* mappings, which comprise a small subset of *competing* transfer mappings (discussed in Section 3.3). For instance, in our test corpus, on average, 38.67 mappings applied to each sentence, of which 12.76 mappings competed with at least one other mapping, but of these only 2.35 were conflicting (and hence had decision trees built for them). We intend on extending this approach to all competing mappings, which is likely to have a much greater impact. This is, however, not entirely straightforward, since such matches compete on some sentences but not others.

In addition, we would like to explore whether abstracting away from specific lemmas, using thesaurus classes, WordNet syn-sets or hypernyms, etc. would result in improved classifier performance.

11 Acknowledgements

Thanks go to Robert C. Moore for many helpful discussions, useful suggestions and advice, particularly in connection with the smoothing method we used. Thanks to Simon Corston-Oliver for advice on decision trees and code to use them, and to Max Chickering, whose excellent tool-kit we used extensively. Thanks also go to members of the NLP group at Microsoft Research for valuable feedback.

References

1. Hiroyuki Kaji, Yuuko Kida, and Yasutsugu Morimoto: Learning Translation Templates from Bilingual Text. In Proceedings of COLING (1992)
2. Adam Meyers, Michiko Kosaka and Ralph Grishman: Chart-based transfer rule application in machine translation. In Proceedings of COLING (2000)
3. Hideo Watanabe, Sado Kurohashi, and Eiji Aramaki: Finding Structural Correspondences from Bilingual Parsed Corpus for Corpus-based Translation. In Proceedings of COLING (2000)
4. Benoit Lavoie, Michael White and Tanya Korelsky: Inducing Lexico-Structural Transfer Rules from Parsed Bi-texts. In Proceedings of the Workshop on Data-driven Machine Translation, ACL 2001, Toulouse, France (2001)
5. Stephen D. Richardson, William Dolan, Monica Corston-Oliver, and Arul Menezes, Overcoming the customization bottleneck using example-based MT. In Proceedings of the Workshop on Data-Driven Machine Translation, ACL 2001, Toulouse, France (2001)
6. Adam Meyers, Roman Yangarber, Ralph Grishman, Catherine Macleod, and Antonio Moreno-Sandoval: Deriving transfer rules from dominance-preserving alignments. In Proceedings of COLING (1998)
7. Hideo Watanabe: A method for distinguishing exceptional and general examples in example-based transfer systems. In Proceedings of COLING (1994)
8. Arul Menezes and Stephen D. Richardson: A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In Proceedings of the Workshop on Data-Driven Machine Translation, ACL 2001, Toulouse, France (2001)
9. David Maxwell Chickering, David Heckerman, and Christopher Meek: A Bayesian approach to learning Bayesian networks with local structure. In D. Geiger, and P. Pundalik Shenoy (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*. 80-89. (1997)
10. Stanley Chen and Joshua Goodman: An empirical study of smoothing techniques for language modeling, In Proceedings of ACL (1996)
11. Frederick Jelinek and Robert L. Mercer: Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands (1980)
12. David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, Carl Kadie: Dependency networks for inference, collaborative filtering and data visualization. In *Journal of Machine Learning Research* 1:49-75 (2000)

Fast and Accurate Sentence Alignment of Bilingual Corpora

Robert C. Moore

Microsoft Research
Redmond, WA 98052, USA
bobmoore@microsoft.com

Abstract. We present a new method for aligning sentences with their translations in a parallel bilingual corpus. Previous approaches have generally been based either on sentence length or word correspondences. Sentence-length-based methods are relatively fast and fairly accurate. Word-correspondence-based methods are generally more accurate but much slower, and usually depend on cognates or a bilingual lexicon. Our method adapts and combines these approaches, achieving high accuracy at a modest computational cost, and requiring no knowledge of the languages or the corpus beyond division into words and sentences.

1 Introduction

Sentence-aligned parallel bilingual corpora have proved very useful for applying machine learning to machine translation, but they usually do not originate in sentence-aligned form. This makes the task of aligning such a corpus of considerable interest, and a number of methods have been developed to solve this problem. Ideally, a sentence-alignment method should be fast, highly accurate, and require no special knowledge about the corpus or the two languages.

Kay and Röscheisen [1][2] developed an iterative relaxation approach to sentence alignment, but it was not efficient enough to apply to large corpora. The first approach shown to be effective at aligning large corpora was based on modeling the relationship between the lengths of sentences that are mutual translations. Similar algorithms based on this idea were developed independently by Brown, et al. [3] and Gale and Church [4][5]. Subsequently, Chen [6] developed a method based on optimizing word-translation probabilities which he showed gave better accuracy than the sentence-length-based approach, but was “tens of times slower than the Brown and Gale algorithms” [6, p. 15]. Wu [7] used a version of Gale and Church’s method adapted to Chinese along with lexical cues in the form of a small corpus-specific bilingual lexicon to improve alignment accuracy in text regions containing multiple sentences of similar length. Melamed [8][9] also developed a method based on word correspondences, for which he reported [8] sentence-alignment accuracy slightly better than Gale and Church. Simard and Plamondon [10] developed a two-pass approach, in which a method similar to Melamed’s identifies points of correspondence in the text that constrain a second-pass search that uses a statistical translation model.

All these prior methods require particular knowledge about the corpus or the languages involved. The length-based methods require no special knowledge about the

languages, but the implementations of Brown et al. and Gale and Church require either corpus-dependent anchor points, or prior alignment of paragraphs to constrain the search. The word-correspondence-based methods of Chen and Melamed do not require this sort of information about the corpus, but they either require an initial bilingual lexicon, or they depend on finding cognates in the two languages to suggest word correspondences. Wu's method also requires that the bilingual lexicon be externally supplied. Simard and Plamondon's approach relies on the existence of cognates for the first pass, and a previously-trained word-translation model for the second pass.

We have developed a hybrid sentence-alignment method, using previous sentence-length-based and word-correspondence-based models, that is fast, very accurate, and requires only that the corpus be separated into words and sentences. In a direct comparison with a length-based model that is a slight modification of Brown et al.'s, we find our hybrid method has a precision error rate 5 to 13 times smaller, and a recall error rate 5 to 38 times smaller. Moreover, the ratio of the computation times required for our method, vs. the length-only-based method, is less than 3 for easy to align material and seems to asymptotically approach 1 as the material becomes harder to align, which is when our advantage in precision and recall is greatest.

2 Description of the Algorithm

Our algorithm combines techniques adapted from previous work on sentence and word alignment in a three-step process. We first align the corpus using a modified version of Brown et al.'s sentence-length-based model. We employ a novel search-pruning technique to efficiently find the sentence pairs that align with highest probability without the use of anchor points or larger previously aligned units. Next, we use the sentence pairs assigned the highest probability of alignment to train a modified version of IBM Translation Model 1 [11]. Finally, we realign the corpus, augmenting the initial alignment model with IBM Model 1, to produce an alignment based both on sentence length and word correspondences. The final search is confined to the minimal alignment segments that were assigned a nonnegligible probability according to the initial alignment model, which reduces the size of the search space so much that this alignment is actually faster than the initial alignment, even though the model is much more expensive to apply to each segment.

Our method is similar to Wu's [7] in that it uses both sentence length and lexical correspondences to derive the final alignment, but since the lexical correspondences are themselves derived automatically, we require no externally supplied lexicon. We discuss each of the steps of our approach in more detail below.

2.1 Sentence-Length-Based Alignment

Brown et al. [3] assume that every parallel corpus can be aligned in terms of a sequence of minimal alignment segments, which they call "beads", in which sentences align 1-to-1, 1-to-2, 2-to-1, 1-to-0, or 0-to-1.¹ The alignment model is a generative probabilistic

¹ This assumption fails occasionally when there is an alignment of 2-to-2 or 3-to-1, etc. This is of little concern, however, because it is sufficient for our purposes to extract the 1-to-1

model for predicting the lengths of the sentences composing sequences of such beads. The model assumes that each bead in the sequence is generated according to a fixed probability distribution over bead types, and for each type of bead there is a submodel that generates the lengths of the sentences composing the bead.

For the 1-to-0 and 0-to-1 bead types, there is only one sentence in each bead, and the lengths of those sentences are assumed to be distributed according to a model based on the observed distribution of sentence lengths in the text in the corresponding language. For all the other beads types (1-to-1, 2-to-1, and 1-to-2), the length(s) of the sentence(s) of the first (source) language are assumed to be distributed according to the same model used in the 1-to-0 case, and the total length of the sentence(s) of the second (target) language in the bead is assumed to be distributed according to a model conditioned on the total length of the sentence(s) of the source language in the bead. Brown et al. assume that the logarithm of the ratios of the length l_t of the sentence(s) of the target language to the length l_s of the corresponding sentence(s) of the source language varies according to a Gaussian distribution with mean μ and variance σ^2 ,

$$P(l_t|l_s) = \alpha \exp(-((\log(l_t/l_s) - \mu)^2/2\sigma^2)), \quad (1)$$

where α is chosen to make $P(l_t|l_s)$ sum to 1 for positive integer values of l_t .

The major difference between our sentence-length-based alignment model and that of Brown et al. is in how the conditional probability $P(l_t|l_s)$ is estimated. Our model assumes that l_t varies according to a Poisson distribution whose mean is simply l_s times the ratio r of the mean length of sentences of the target language to the mean length of sentences of the source language:

$$P(l_t|l_s) = \exp(-l_s r) (l_s r)^{l_t} / (l_t!). \quad (2)$$

The idea is that each word of the source language translates into some number of words in the target language according to a Poisson distribution, whose mean can be estimated simply as the ratio of the mean sentence lengths in the two languages. This model is simple to estimate because it has no hidden parameters, whereas at least the variance σ^2 needs to be estimated iteratively using EM in Brown et al.'s Gaussian model. Moreover, when we compared the two models on several thousand sentences of hand-aligned data, we found that the Poisson distribution actually fit the data slightly better than the best-fitting Gaussian distribution of the form used by Brown et al.

There are a few other minor differences between the two models. Brown et al. estimate marginal distributions of sentence lengths in the two languages using the raw relative frequencies in the corpus to estimate the probabilities of the lengths of shorter sentences, and smooth the estimates for the lengths of longer sentences by fitting to the tail of a Poisson distribution. In contrast, we simply use the raw relative frequencies to estimate the probability of every observed sentence length. This only affects the estimates for particularly long, and therefore rare, sentence lengths, which should have no appreciable effect on the performance of the model. We also found that the performance of the model was rather insensitive to the exact values of the probabilities assigned to

alignments, which account for the vast majority of most parallel corpora and are in practice the only alignments that are currently used for training machine translation systems.

the various bead types, so we simply chose rough values close to those reported by Brown et al. and Gale and Church, rather than tuning them by re-estimation as Brown et al. do. We experimented with initializing the model with these values and iteratively re-estimating to the optimal values for our data, but we never saw a significant difference in the output of alignment as a result of re-estimating these parameters. Finally, Brown et al. also include paragraph boundary beads in their model, which we omit, in part because paragraph boundary information was not present in our data.

Our intention in making these modifications to the model of Brown et al. is not to improve its accuracy in sentence alignment, and we certainly do not claim to have done so. In fact, we believe that the differences are so slight that the models should perform comparably. The practical difference between the two models is that because ours has no hidden parameters, we don't need to use EM or any other iterative parameter re-estimation method, which makes our variant much faster to use in practice.

Search Issues The standard approach to solving alignment problems is to use dynamic programming (DP). In an exhaustive DP alignment search, one iteratively computes some sort of cost function for all possible points of correspondence between the two sequences to be aligned. For the sentence alignment problem, the number of such points is approximately the product of the numbers of sentences in each language; so it is clearly infeasible to do an exhaustive DP search for a large corpus. The search must therefore be pruned in some way, which is the approach we have followed, as have Brown et al., Gale and Church, and Chen. Our method of pruning, however, is novel and has proved quite effective.

Notice that unless there are extended segments of one language not corresponding to anything in the other language, the true points of correspondence should all be close to proportionately the same distance from the beginning of each text. For example, the only way a point 30% of the way along the text in the source language would be likely to correspond to a point 70% of the way along the text in the target language is if there were some major insertions and/or deletions in one or both of the texts. Following Melamed, we think of the set of possible points of correspondence as forming a matrix, and the set of points closest to proportionately the same distance from the beginning of each text as “the main diagonal”.

Our pruned DP search starts by doing an exhaustive search, but only in a narrow fixed-width band around the main diagonal. Unless there are extended segments of one language not corresponding to anything in the other language, the best alignment of the two texts will usually fall within this band. But how do we know whether this is the case? Our heuristic is to look at an approximate best alignment within the band, and find the point where it comes closest to one of the boundaries of the band. If the approximate best alignment never comes closer than a certain minimum distance from the boundaries of the band, we assume that the best alignment within the band is actually the best possible alignment, and the search terminates. Otherwise, we widen the band and iterate. While we have no proof that this heuristic will always work, we have never seen it commit a search error in practice. Our conjecture is that if the search band is too narrow to contain the true best alignment, the constrained best alignment will basically be a random walk in those regions where the true best alignment is excluded. If the

size of the excluded regions of the true best alignment is large, the probability of this random walk never coming close to a boundary is small.

In this phase of our algorithm, the main goal is to find all the high-probability 1-to-1 beads to use for training a word-translation model. We find these beads by performing the forward-backward probability computation, as described by Rabiner [12], using the initial search described above as the forward pass. To speed up the backward pass of this search, we start by considering only points that have survived the first pass pruning, and we further prune out (and do not extend) any of these points that receive a very low total probability in the backward pass.

2.2 Word-Translation Model

In the next phase of our algorithm, we use the highest probability 1-to-1 beads from the initial alignment to train a word-translation model. We use a threshold of 0.99 probability of correct alignment to ensure reliable training data, and in our experiments this makes use of at least 80% of the corpus. For our word-translation model, we use a modified version of the well-known IBM Translation Model 1 [11].

The general picture of how a target language sentence t is generated from a source language sentence s consisting of l words, $s_1 \dots s_l$, in the IBM translation models is as follows: First, a length m is selected for t . Next, for each word position in t , a generating word in s (including the null word s_0) is selected. Finally, for each pair of a position in t and its generating word in s , a target language word is chosen to fill the target position. Model 1 makes the assumptions that all possible lengths for t (less than some arbitrary upper bound) have a uniform probability ϵ ; all possible choices of the source language generating words are equally likely; and the probability $tr(t_j|s_i)$ of the generated target language word depends only on the generating source language word—which Brown et al. show yields

$$P(t|s) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^m \sum_{i=0}^l tr(t_j|s_i). \quad (3)$$

We make two minor modifications in Model 1 for the sake of space efficiency. The translation probabilities for rare words can be omitted without much loss, since they will hardly ever be used. Therefore, to prune the size of our word-translation model, we choose a minimum number of occurrences for a word to be represented distinctly in the model, and map all words with fewer occurrences into a single token prior to computing the word-translation model. For each language, we set the threshold to be the maximum count per word that is sufficient to result in 5000 distinct words of that language, subject to an absolute minimum for the threshold of 2 occurrences per word.

In principle, Model 1 will assign a translation probability to every possible pair consisting of one of the remaining words from each language, provided the words both occur in at least one aligned sentence pair. The vast majority of these, however, will not represent true translation pairs and therefore contribute little to determining correct sentence alignment. Therefore, our second modification to Model 1 is that, in accumulating fractional counts in each iteration of EM after the first, any fractional count for a word-translation pair in a given sentence that is not greater than would be obtained by making a totally random choice is not added to the count for that translation pair.

For example, if a source language sentence contains 10 words (including the null word) and the existing model assigns to one of those words a fractional count not greater than 0.1 for generating a particular word in the target language sentence, we don't include that fractional count in the total count for that word-translation pair. To maintain the integrity of the model, we assign these fractional counts to the pair involving the null word instead. We find this reduces the size of the model by close to 90% without significantly impacting the performance of the resulting model.

We train our modified version of Model 1 by carrying out 4 iterations of EM as described by Brown, et al. [11], which we found to be an upper bound on the number of iterations needed to minimize the entropy of held out data.

2.3 Word-Correspondence-Based Alignment

For the final sentence-alignment model we use the framework of our initial sentence-length-based model, but we modify it to use IBM Model 1 in addition to the initial model. The modified model assumes that bead types and sentence lengths are generated according to the same probability distributions used by the sentence-length-based model, but we multiply the probability estimate based on these features by an estimated probability for the actual word sequences composing each bead, based on the instance of Model 1 that we have estimated from the initial alignment.

For the single sentence in a 1-to-0 or 0-to-1 bead, each word is assumed to be generated independently according to the observed relative unigram frequency f_u of the word in the text in the corresponding language. For all the other beads types (1-to-1, 2-to-1, and 1-to-2), the words of the sentence(s) of the source language are assumed to be generated according to the same model used in the 1-to-0 case; and the words of the sentence(s) of the target language in the bead are assumed to be generated depending on the words of the source language, according to the instance of Model 1 that we have estimated from the initial alignment of the corpus. In applying Model 1, we omit the factor corresponding to the assumption of uniform distribution of target sentence lengths, since we have already accounted for sentence length by incorporating our original alignment model. For example, if s is a source sentence of length l , t is a target sentence of length m , and $P_{1-1}(l, m)$ is the probability assigned by the initial model to a sentence of length l aligning 1-to-1 with a sentence of length m , then our combined model will estimate the probability of a 1-to-1 bead consisting of s and t as

$$P(s, t) = \frac{P_{1-1}(l, m)}{(l+1)^m} \left(\prod_{j=1}^m \sum_{i=0}^l tr(t_j | s_i) \right) \left(\prod_{i=1}^l f_u(s_i) \right). \quad (4)$$

Simard and Plamondon [10] also base their second pass on IBM Model 1. However, because they essentially use *only* Model 1—without embedding it in a more general framework, as we do—they have no way to assign probabilities to 1-to-0 and 0-to-1 beads. Hence their model has no way to accomodate deletions or insertions, which they conjecture results in the low precision they observe on many of their test corpora [10, p. 77].

Since our hybrid alignment model incorporating IBM Model 1 is much more expensive to apply to a bead than our original sentence-length-based model, if we were

Table 1. Results for Manual 1 data

Alignment Method	Probability Threshold	Number Right	Number Wrong	Number Omitted	Precision Error	Recall Error
Hand-Aligned	NA	9842	1	6	0.010%	0.061%
Length Only	0.5	9832	28	16	0.284%	0.162%
Length+Words	0.5	9846	5	2	0.051%	0.020%
Length+Words	0.9	9839	3	9	0.030%	0.091%

Table 2. Results for Manual 2 data

Alignment Method	Probability Threshold	Number Right	Number Wrong	Number Omitted	Precision Error	Recall Error
Hand-Aligned	NA	17276	5	99	0.029%	0.570%
Length Only	0.5	17304	18	71	0.104%	0.409%
Length+Words	0.5	17361	2	14	0.012%	0.081%
Length+Words	0.9	17316	1	59	0.006%	0.340%

to start the alignment search over from scratch, generating the final alignment would be very slow. We limit the search, however, to the set of possible points of correspondence receiving nonnegligible probability estimates in the initial sentence-length-based alignment. Since these are only a small fraction (on the order of 10% or less) of all the possible points correspondence explored in the initial alignment search, this greatly speeds up the final alignment search. In practice, the final alignment search takes less time than the initial alignment search—far less in some cases.

3 Results

We have evaluated our method on data from two English-language computer software manuals and their Spanish translations, for which we were able to obtain hand alignments of 1-to-1 beads for comparison. The automatic and hand alignments were in close enough agreement that we were able to have all the differences examined by a fluent bilingual. In some cases we found the hand alignment to be in error and the automatic alignment to be correct. For the purposes of our analysis we assume that every alignment pair that the automatic and hand alignments agree on is correct, and that all the correct alignment pairs are found either by the hand alignment or automatic alignment.

Our evaluation metrics are precision error and recall error for 1-to-1 sentence alignments. We follow Brown et al. [3, pp. 175–176] in using precision error (which they simply call “error”) on 1-to-1 beads (which they call “*ef*-beads”) as an evaluation metric. Because we have complete hand alignments for the 1-to-1 beads for all our test data, however, we are also able to measure recall error, which many previous studies have not been able to estimate.

Our results on data from Manual 1 are shown in Table 1, and results from Manual 2 are shown in Table 2. For each manual, we compare results for four different alignments: hand alignment, alignment based on sentence length only at the 0.5 probability

threshold, alignment based on sentence length and word correspondence at the 0.5 probability threshold, and alignment based on sentence length and word correspondence at the 0.9 probability threshold. The probability threshold refers to a cut-off based on the probability assigned to an alignment by application of the forward-backward probability computation as discussed in Section 2.1. Since we are able to estimate this probability, rather than simply computing the most probable overall alignment sequence, we can tune the precision/recall trade-off depending on where we decide to set our threshold for accepting an alignment pair.

Examining the results in Tables 1 and 2 shows that both the precision and recall error rates for all alignments are well under 1.0%, but that recall error and precision error are considerably lower for our hybrid model than for the alignment based only on sentence length. At the probability threshold of 0.5, for Manual 1 the precision error was 5.6 times lower and the recall error was 8.0 times lower for the hybrid method, and for Manual 2 the precision error was 9.0 times lower and the recall error was 5.1 times lower for the hybrid method. For Manual 1 the precision and recall error for the hybrid method (at either the 0.5 or 0.9 probability threshold) were almost as good as on the hand-aligned data, and for Manual 2 the error rates were actually better for data aligned by the hybrid method than for the hand-aligned data.

We believe that the data we have used in these experiments is representative of much of the sort of parallel data one might encounter as training data for machine translation. However, it turns out to be fairly easy data to align, as indicated by the low error rates of both forms of automatic alignment that we applied, and by the fact that the highest probability initial alignments deviated from the main diagonal by at most 6 positions in the case of the data from Manual 1 and at most 13 positions in the case of the data from Manual 2. To test how well the algorithms perform on more difficult data, we applied both the method based only on sentence length and the hybrid method to versions of the Manual 1 data, from which single blocks of 50, 100, and 300 sentences had been deleted from one side of the corpus at a randomly chosen point.

The results of this experiment are shown in Table 3, for the 0.5 probability threshold. Examining these results shows that as the size of the deletion increases, the precision and recall error rates for the alignment based only on sentence length also increase, but the error rates for hybrid method remain essentially constant. The advantage of the hybrid method thus increases to the point that, on the data with 300 sentences deleted, the precision error is 13.0 times lower and the recall error is 37.4 times lower than with the sentence-length-only-based method.

These substantial deletions stress the search strategy as well as the alignment models, since they force the initial search to examine a much wider band around the main diagonal to find the optimal alignment. We show the effect on the total time to compute the alignments in Table 4. Of necessity, the forward pass time of the sentence-length-only-based alignment increases at least in proportion to the maximum deviation of the best alignment from the main diagonal. If the width of the search band is doubled on every iteration, then the total search time should be no more than twice the time of the last iteration, and the width of the search band should be no more than twice the maximum deviation of the best alignment from the main diagonal. This means it should be possible to carry out the iterative first pass search in time proportional to the length of

Table 3. Results for Manual 1 data with deletions

Sentences Deleted	Alignment Method	Number Right	Number Wrong	Number Omitted	Precision Error	Recall Error
0	Length Only	9832	28	16	0.284%	0.162%
50	Length Only	9761	30	39	0.306%	0.398%
100	Length Only	9677	30	73	0.309%	0.749%
300	Length Only	9368	52	187	0.552%	1.967%
0	Length+Words	9846	5	2	0.051%	0.020%
50	Length+Words	9796	6	4	0.061%	0.041%
100	Length+Words	9747	5	3	0.051%	0.031%
300	Length+Words	9550	4	5	0.042%	0.052%

Table 4. Alignment time (in seconds) for deletion experiments

Sentences Deleted	First Pass Iterations	Length Align Time	Model 1 Train Time	Length+Words Align Time	total Total
0	1	161	131	155	447
50	3	686	133	195	1013
100	5	1884	128	281	2293
300	7	4360	125	555	5040

the corpus times the size of the maximum deviation of the best alignment from the main diagonal. This seems roughly consistent with the increasing times for sentence-length-only-based alignment as the number of sentences deleted goes from 50 to 300.

Naturally, the time to train IBM Model 1 is essentially independent of the difficulty of the initial alignment. What is particularly striking, however, is that the time to perform the final alignment goes up much more slowly than the time to perform the initial alignment, due to its restriction to evaluation of points of alignment receiving nonnegligible probability in the initial alignment. As the difficulty of the alignment task increases in these experiments, the ratio of the time to perform the complete alignment process to the time to perform the initial alignment decreases from 2.8 to 1.2, with every indication that it should asymptotically approach 1.0. Thus for difficult alignment tasks, we gain the error reduction of the hybrid method at almost no additional relative cost.

4 Conclusions

It was perhaps first shown by Chen [6] that word-correspondence-based models can be used to produce higher-accuracy sentence alignment than sentence-length-based models alone. The main contribution of this work is to show how get the benefit of those higher accuracy models with only a modest additional computational cost, and without the use of anchor points, cognates, a bilingual lexicon—or any other knowledge about the corpus other than its division into words and sentences. In accomplishing this, we have made the following novel contributions to the statistical models and the search strategies used:

1. Modification of Brown et al.'s [3] sentence-length-based model to use Poisson distributions, rather than Gaussians, so that no hidden parameters need to be iteratively re-estimated.
2. A novel iterative-widening search method for alignment problems, based on detecting when the current best alignment comes near the edge of the search band, which eliminates the need for anchor points.
3. Modification of IBM Translation Model 1, eliminating rare words and low probability translations to reduce the size of the model by 90% or more.
4. Use of the probabilities computed by a relatively cheap initial model (the sentence-length-based model) to dramatically reduce the search space explored by a second more accurate, but more expensive model (the word-correspondence-based model). While this idea has been used in such fields as speech-recognition and parsing, it seems not to have been used before in bilingual alignment.

References

1. Kay, M., Röscheisen, M.: Text-Translation Alignment. Technical Report, Xerox Palo Alto Research Center (1988)
2. Kay, M., Röscheisen, M.: Text-Translation Alignment. *Computational Linguistics* **19**(1) (1993) 121–142
3. Brown, P.F., Lai, J.C., Mercer, R.L.: Aligning Sentences in Parallel Corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California (1991) 169–176
4. Gale, W.A., Church, K.W.: A program for Aligning Sentences in Bilingual Corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California (1991) 177–184
5. Gale, W.A., Church, K.W.: A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics* **19**(1) (1993) 75–102
6. Chen, S.F.: 1993. Aligning Sentences in Bilingual Corpora Using Lexical Information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio (1993) 9–16
7. Wu, D.: Aligning a Parallel English-Chinese Corpus Statistically with Lexical Criteria. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico (1994) 80–87
8. Melamed, I.D.: A Geometric Approach to Mapping Bitext Correspondence. IRCS Technical Report 96-22, University of Pennsylvania (1996)
9. Melamed, I.D.: A Portable Algorithm for Mapping Bitext Correspondence. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain (1997) 305–312
10. Simard, M., Plamondon, P.: Bilingual Sentence Alignment: Balancing Robustness and Accuracy. *Machine Translation* **13**(1) (1998) 59–80
11. Brown, P.F., Della Pietra, S. A., Della Pietra, V. J., Mercer, R.L.: The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* **19**(2) (1993) 263–311
12. Rabiner, L. R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* **77**(2) (1989) 257–286

Deriving Semantic Knowledge from Descriptive Texts Using an MT System

Eric Nyberg¹, Teruko Mitamura¹, Kathryn Baker¹,
David Svoboda¹, Brian Peterson², and Jennifer Williams²

¹ Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{ehn,teruko,klb,svoboda}@cs.cmu.edu

² Ontology Works, Inc.
1132 Annapolis Rd, Suite 104
Odenton, MD 21113-1672
{peterson,williams}@ontologyworks.com

Abstract. This paper describes the results of a feasibility study which focused on deriving semantic networks from descriptive texts using controlled language. The KANT system [3,6] was used to analyze input paragraphs, producing sentence-level interlingua representations. The interlinguas were merged to construct a paragraph-level representation, which was used to create a semantic network in Conceptual Graph (CG) [1] format. The interlinguas are also translated (using the KANTOO generator) into OWL statements for entry into the Ontology Works electrical power factbase [9]. The system was extended to allow simple querying in natural language.

1 Introduction

This paper reports on a study which adapted machine translation tools to generate knowledge representation languages as part of a knowledge extraction system. The inputs to the system are short texts describing critical infrastructures (e.g. financial markets, electrical power transmission). Where necessary, the texts are rewritten to conform to a controlled language [4]. Then each text is analyzed to produce an interlingua representation (IR) for its sentences. The interlinguas are merged together into a single representation for the paragraph. The merged representation is then generated into different output languages. In this study, the outputs were not generated in natural language, but as statements in different knowledge representation languages - Conceptual Graphs (CG) / Knowledge Interchange Format (KIF) [1], and OWL [9] (see Figure 1).

We used the KANTOO MT system [7] for the analysis and generation steps. We investigated two output representations: Conceptual Graph / Knowledge Interchange Format (KIF) [1], and the Ontology Works OWL language [9]. The

study focused on an ontology and textual description for a model of the North-west electric power grid [10]. A set of texts were written to describe the elements of the model and their attributes; these texts were re-written to conform to a controlled language. The KANTOO analyzer was extended to produce merged interlingua structures for these texts, and the KANTOO generator was extended with special knowledge for generating from merged interlinguas into KIF and OWL. The OWL statements were instantiated as facts in an Ontology Works factbase, instantiating concepts in an upper model for the electric power domain.

In an extension of the basic system, we enhanced the KANTOO analyzer with the ability to process natural language questions about the concepts in the factbase. After translating these questions into the appropriate OWL queries, KANTOO was able to query the factbase and return the appropriate results.

In Section 2 we describe the controlled language analysis used to create interlinguas for the input sentences. Section 3 outlines the merging algorithm used to create semantic graphs from the interlinguas. In Section 4 we discuss the generation of KIF and OWL output from the interlingua. Section 5 discusses an initial capability for natural language querying of the resulting fact base. We conclude in Section 6 with some remarks about open issues and possible future work.

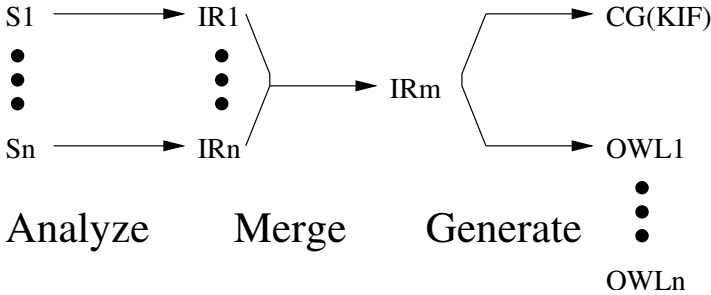


Fig. 1. Three-Step Processing

2 Controlled Language Analysis

The general goals of Controlled Language (CL) are to achieve consistent authoring of source texts and to encourage clear and direct writing. Controlled language is also used to improve the quality of translation output through the use of short, concise and unambiguous sentences [4,2,8]. Although the goal of this study was to generate knowledge representations, rather than translated texts, this task also requires an accurate, unambiguous meaning representation.

2.1 Controlled Language in KANT

In KANT Controlled English (KCE) we explicitly encode domain lexemes along with their meanings. Whenever possible, the lexicon should encode a single mean-

ing for each word/part-of-speech pair. When a term must carry more than one meaning in the domain, these meanings must be encoded in a separate lexical entry. If more than one such entry is activated during analysis, and the ambiguity cannot be resolved by the grammar, then interactive disambiguation is used to select the intended meaning. The analysis grammar encodes a set of writing guidelines and constraints which reduce ambiguity, both at the phrasal level and at the sentential level (for more details, see [4]).

2.2 The KANT Interlingua

The KANT analyzer uses a lexicon and a unification grammar to produce a set of LFG-style f-structures. These f-structures are mapping into interlingua expressions by a semantic interpreter module. Each interlingua may contain semantic concepts, semantic features, and semantic roles representing the meaning of various lexical and structural components of the input sentence. Some grammatical information may also be preserved in the interlingua, if it is necessary for accurate translation. An example interlingua is shown in Figure 2.

```
(*A-DETERMINE
(argument-class agent+patient)
(mood imperative)
(patient
  (*O-VULNERABILITY
    (distance near)
    (number mass)
    (q-modifier
      (*Q-characteristic_WITH
        (case
          (*K-WITH))
        (object
          (*O-PB-TRADER
            (number plural)
            (reference no-reference)))
          (role characteristic)))
        (reference definite)))
    (punctuation period)
    (tense present))
```

Fig. 2. KANT Interlingua for *Determine this vulnerability with PB traders*

In KANT, the concept is lexically specified by the head of the syntactic constituent that the interlingua corresponds to. A semantic role is a slot that is filled with an embedded interlingua expression. The embedded interlingua is headed by the concept associated with the head of the syntactic constituent that the semantic role corresponds to. For example, **subject** and **object** in the f-structure can correspond to **agent** and **theme** in the interlingua.

2.3 Designing a Controlled Language for Descriptive Texts

One primary objective of the study was to design a CL grammar whose interlingua output could be mapped into knowledge representation languages. The first step was to determine a set of paragraph-level writing guidelines, by rewriting sample texts into a form suitable for automatic analysis. The rewriting strategies we used can be thought of as an initial specification for CL authoring at the paragraph level. This is a significant departure from past work in CL, where the unit of analysis is typically a sentence. The sample texts were drawn from two different domains: economic system vulnerability and power system vulnerability (see Figure 3). The first step in applying a CL to a new domain is to add unfamiliar terms to the lexicon. We used our existing English lexicon and added new terms from the economic and power domains.

Original:

Vulnerability of the economic system Sigma sub H2 was studied by determining the U-vulnerability and Lambda-vulnerability of a few simple system instantiations.

Rewritten:

Determine U-vulnerability and Lambda-vulnerability of a few simple instantiations of Sigma sub H2, which is an economic system.

Original:

The U-vulnerability for an ST exchange rate model instantiation of Sigma sub H2 was examined for the case in which the currency trader estimate of "true" exchange rate is manipulated. For this instantiation, and with "output" defined to be GDP volatility magnitude, U-vulnerability was determined using signal bounding arguments.

Rewritten:

The U-vulnerability of the ST exchange rate model Sigma sub H2 was determined with the output as GDP volatility magnitude. This U-vulnerability was determined by manipulation of the trader estimate of the true exchange rate. This U-vulnerability was determined by using signal bounding arguments.

Fig. 3. Example Texts Rewritten to CL

We experimented with two different approaches to rewriting the source texts. First, we took the same approach that we have used for past domains, where an existing text is rewritten on a sentence by sentence basis, according to the KANT CL Specification. This method focuses on grammatical constructions. We found that while this approach was not particularly difficult, the interlingua output did not necessarily model the semantic relations required to build the graph structures required for the CG output. To reduce the complexity of mapping interlingua into CG format, it is necessary to promote isomorphism, in the sense that meanings which are represented with identical structures in CG should be represented using parallel structures in the KCE input and interlingua.

The second approach to rewriting was to define the simplest canonical sentence structure for each graph relation produced in the first approach, working

backward from graph to text. The original English sentences were used only for clarification or verification of meaning. This novel method of working from graph to text is central to the success of the study. This method has the advantage of addressing several sentences at once, because a large graph conveys the meaning of more than one sentence at a time. Graphs need not be tied to a sentence-level analysis. Another advantage is that we were able to produce simple but meaningful sentences, even though our researchers were unfamiliar with the domain. Non-domain experts should find the process accessible.

For these reasons, we decided to use this graph-based approach to rewriting for the rest of the study. We continued to use sample graphs as a primary reference in designing the CL sentence structures.

2.4 Refining the CL for Paragraphs

We began to look at shared elements from sentence to sentence, in order to reason about how the interlinguas from two or more sentences might be merged. We established some initial guidelines for structuring the text around CG creation. In the field of CL generally, and also for graph-based CL, it is essential to follow the guidelines consistently in order to obtain the best possible output.

One guideline is to repeat the same verb and object from a single graph with different modifiers. For example, if the graph is headed by the action *Determine* and the theme is *Lambda-vulnerability*, then the phrase *Determine Lambda-vulnerability* can be repeated with the **Chrc** (characteristic) *MM & MA traders* and with the **Rslt** (result) *output as GDP volatility magnitude*.

- **(Before)** Determine this Lambda vulnerability *with MM traders and MA traders*.
- **(After)** Determine this Lambda vulnerability *with output as GDP volatility magnitude*.

Another writing guideline requires the writer to refer to a mass noun, such as *U-vulnerability*, by using a determiner such as *the* or *this*, once the noun has been introduced into the context:

- **(Before)** U-vulnerability of the ST exchange rate model Sigma sub H2 was determined with the output as GDP volatility magnitude.
- **(After)** *This U-vulnerability* was determined by manipulation of the trader estimate of the true exchange rate.

Other ongoing work includes developing a set of *key words*. Key words written in the text can correspond to graph node creation. For example, *in order to* signals the graph node **Purpose**. The phrase *By using* signals the graph node **Method**. Use of key words is important because it enables directed writing by the authors and also makes the graphs more intuitive and clear.

Once a set of IRs is produced, the goal is to combine them into a single IR structure that corresponds to a unique graph. Features in an IR which were superfluous for this goal, such as syntactic features which would be used by a

translation system, were identified as features which could be omitted. As patterns for merging were identified, these were factored into the merging algorithm, which will be described in the following section.

3 Merging Interlinguas into CG

The interlingua merging algorithm produces a single conceptual graph (CG) output for the set of interlinguas (IRs) produced by KANT for a particular text. In this section, we describe the steps that are taken to produce the merged IRs:

1. **Preprocess the IRs.** This entails removing some structural slots that exist in the Interlingua Representations (IRs) that aren't necessary in order to construct CG output. For example, certain grouping structures such as *G-COORDINATION are replaced with an ordered list of slot fillers.
2. **Construct an 'IR Index' for Every Concept.** Indexing each concept is an important step towards finding similar IRs to merge into one in the later step. The means of finding the concept is represented using a path of slots. We can use this index to locate and compare a set of sub-IRs that all start with the same concept.
3. **Find Concepts That Appear In More Than One IR.** To merge the IRs, we must identify IRs which can be unified; i.e., their meaning are consistent and can be combined into a single graph. A concept has multiple instances if it heads two or more sub-IRs that cannot be unified. These sub-IRs can be determined using the IR Index generated in Step 2. If all the sub-IRs for a concept unify, the concept has only one unique instance. Otherwise, the number of instances of the concept is equal to the minimum number of sub-interlinguas that cannot be unified.

This simple type of unification (via concept equivalence) might be extended to unification under subsumption, or by merging concepts using coordination; e.g., the concepts *O- TRADER- ESTIMATE and *O- TRADER- EXCHANGE- RATE- ESTIMATE can be unified rather than treated as separate instances of different concepts.

4. **Create Unique Identifiers for Multiple Instances.** Any remaining instances of the same concept (e.g. *O-MANIPULATION) are re-labelled with unique indices, e.g. *O-MANIPULATION-a and *O-MANIPULATION-b.
5. **Create a Unified IR For Each Concept.** At this point, we can guarantee that each concept identifier used in the IRs has exactly one instance. We associate with each concept identifier an IR that is the unification of all sub-IRs that modify that concept. Since we store each unified IR separately, we can replace any occurrence of an IR as slot filler in another IR with a simple reference (pointer).
6. **Merge The Unified IRs.** At this point we traverse the unified IRs, and substitute other unified IRs for lone concept indices (pointers).
7. **Remove Slots That Don't Contain Concepts.** Remaining features which don't contain concepts (e.g., number, reference) are removed.

The merging algorithm returns all the IRs generated by following these steps in sequence. In the next section, we discuss how the merged IR is generated into KIF and OWL.

4 Generating Knowledge Representations: KIF and OWL

The structures created by the IR merging process are generated as KIF and OWL. Since the two representations are different, we address each in turn. The merged interlingua expressions can be directly generated into CG/KIF form by associating a node with each concept, and an arc with each semantic relation. The merged interlingua expressions are isomorphic to KIF structures, with the exception of certain transformations which are required to map multiply-filled slots in the IR to multiple edges in the KIF graph. The KIF file format utilized also requires that each KIF graph be output on a single line. Due to the inherent near-isomorphism, it was relatively simple to generate KIF output from the merged IR. The generation of KIF structures from IR is completely automatic; the semantic relations in the KIF graph are taken directly from the semantic roles and features in the IR. An example of a completed KIF output can be seen in Figure 4.

```
(*PROP-RAVER
  (comparison-theme
    (*O-COMBUSTION-POWER-PLANT
      (located_IN
        (*PROP-PACIFIC-NORTHWEST-POWER-GRID
          (comparison-theme
            (*O-POWER-GRID
              (attribute (*P-ELECTRICAL)))))))
    (possesses
      (*O-GENERATION-OUTPUT-e
        (attribute (*P-MAXIMAL))
        (value_OF
          (*U-MEGAWATT-e (quantity (''2200'))))))))
```

Fig. 4. Example KIF Graph (ASCII format)

The Ontology Works fact base is implemented as a relational database, with an API that allows semantic predicates and relations to be instantiated via a knowledge representation language called OWL [9]. In order to generate merged IRs as OWL statements, each IR was decomposed into the appropriate set of semantic primitives. The merged interlinguas for the entire text were generated into OWL form, using the KANTOO generator module. These OWL statements

are then passed to the Ontology Works batch loader for insertion into the fact base. An example OWL output for the IR in Figure 4 is shown in Figure 5.

```
(EPctx.EPGrid PNWgrid)
(EPctx.CombustionPowerPlant RAVER)
(EPctx.epPartOf RAVER PNWgrid)
(EPctx.maxRatedGenerationOutput RAVER (MetricCtx.megawatt 2200))
```

Fig. 5. Example OWL Output

5 Querying the Fact Base in Natural Language

Once the semantic information has been extracted from the source text and loaded into the Ontology Works fact base, various queries can be run against the fact base to examine the information that was extracted. OWL includes a query language which can be used to formulate database queries. The initial version of our system can also accept natural language queries, which are mapped to interlingua form by the KANTOO analyzer, and generated as OWL queries by the KANTOO generator. A sample text is shown in Figure 6, and some example queries are shown in Figure 7.

The Pacific Northwest power grid (PNW power grid) is an electrical power grid. Custer, Monroe, Paul, Allston and Keeler are thermal power plants in the PNW power grid. Custer, Monroe, Paul, Allston and Keeler have a maximal generation output of 650 megawatts.

Raver is a combustion power plant in the PNW power grid. The Dalles is a hydro power plant in the PNW power grid. Raver has a maximal generation output of 2200 megawatts. The Dalles has a maximal generation output of 1807 megawatts.

Grand Coulee and Chief Joseph are power plants in the PNW power grid. Grand Coulee and Chief Joseph are members of the Upper Columbia power plants group. The Upper Columbia power plants group is part of the PNW power grid. Grand Coulee has a maximal generation output of 6480 megawatts. Chief Joseph has a maximal generation output of 2520 megawatts.

The Lower Columbia power plants group is part of the PNW power grid. John Day and The Dalles are part of the Lower Columbia power plants group. John Day has a maximal generation output of 1160 megawatts.

Fig. 6. Input Text: Northwest Power Grid Plants

What are the subparts of the PNW power grid?

(EPctx.epPartOf ?x PNWgrid)

Results:

MALIN-ROUND-MOUNTAIN-POWER-LINE-a
BIG-EDDY-THE-DALLES-POWER-LINE-a
CUSTER-MONROE-POWER-LINE-a
PAUL
RAVER
MALIN-ROUND-MOUNTAIN-POWER-LINE-b
ALLSTON
UPPER-COLUMBIA-POWER-PLANTS-GROUP
CHIEF-JOSEPH
CUSTER
THE-DALLES
GRAND-COULEE
KEELER
BIG-EDDY-THE-DALLES-POWER-LINE-b
LOWER-COLUMBIA-POWER-PLANTS-GROUP
MONROE
CUSTER-MONROE-POWER-LINE-b

Are there any hydro power plants?

(EPctx.HydroPowerPlant ?x)

Results:

JOHN-DAY
THE-DALLES

The Dalles is connected to what?

(EPctx.epConnectsTo THE-DALLES ?x)

Results:

BIG-EDDY-THE-DALLES-POWER-LINE-b
BIG-EDDY-THE-DALLES-POWER-LINE-a

Fig. 7. Sample NL Queries, OWL and System Output

6 Conclusion

In this paper we have shown how an existing machine translation system can be adapted to generate output in a knowledge representation language instead of a human language. By combining this capability with a controlled language definition for specific domains (e.g., economic systems, power grids), it is feasible to create knowledge acquisition systems to populate a fact base from natural language texts. The KANTOO machine translation system was used to generate both KIF output structures and OWL statements to represent the facts in texts about economic systems and power grids.

The texts we have examined initially are simple descriptions of the static structures and relationships in the two domains. The system could be used to build and extend the ontology currently used by KANTOO for source text analysis. Most of the semantic knowledge used by the Analyzer is in the form of slot-filler restrictions [5], which could be learned by extracting relevant KIF fragments from texts as they are written for a new domain.

Future work should also focus on dynamic (non-monotonic) descriptions which require more reasoning (i.e., truth maintenance, conflict resolution) in the fact base as what is stated about the domain changes over time.

References

1. Hayes, P. and C. Menzel: A Semantics for the Knowledge Interchange Format, IJCAI 2001 Workshop on the IEEE Standard Upper Ontology, Aug. 6. (2001) 145, 145, 145
2. Kamprath, C., Adolphson, E., Mitamura, T. and Nyberg, E.: Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English. In: Proceedings of the Second International Workshop on Controlled Language Applications (1998) 146
3. Mitamura, T., Nyberg, E. and Carbonell, J.: An Efficient Interlingua Translation System for Multi-lingual Document Production. In: Proceedings of the Third Machine Translation Summit (1991) 145
4. Mitamura, T. and Nyberg, E.: Controlled English for Knowledge-Based MT: Experience with the KANT System. In: Proceedings of TMI-95 (1995) 145, 146, 147
5. Mitamura, T., Nyberg, E., Torrejon E. and Igo, R.: Multiple Strategies for Automatic Disambiguation in Technical Translation, Proceedings of TMI-99 (1999). 154
6. Nyberg, E. and Mitamura, T.: The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains. In: Proceedings of COLING-92 (1992) 145
7. Nyberg, E. and T. Mitamura: The KANTOO Machine Translation Environment. In Proceedings of AMTA-2000. 145
8. Nyberg, E., T. Mitamura and W. Huijsen (to appear). "Controlled Language," in H. Somers, ed., *Computers and Translation: Handbook for Translators*, to be published by Johns Benjamins. 146
9. OWL and the IODE: The Ontology Works White Paper. Available at <http://www.ontologyworks.com/whitepaper.pdf>. August (2001). 145, 145, 145, 151
10. Kosterev, D. N., C. W. Taylor and W. A. Mittelstadt: Model Validation for the August 10th, 1996 WSCC System Outage, IEEE Transactions on Power Systems, Volume 14, Number 03, August (1999). 146

Using a Large Monolingual Corpus to Improve Translation Accuracy

Radu Soricut, Kevin Knight, and Daniel Marcu

Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
{radu, knight, marcu}@isi.edu

Abstract. The existence of a phrase in a large monolingual corpus is very useful information, and so is its frequency. We introduce an alternative approach to automatic translation of phrases/sentences that operationalizes this observation. We use a statistical machine translation system to produce alternative translations and a large monolingual corpus to (re)rank these translations. Our results show that this combination yields better translations, especially when translating out-of-domain phrases/sentences. Our approach can be also used to automatically construct parallel corpora from monolingual resources.

1 Introduction

Corpus-based approaches to machine translation usually begin with a bilingual training corpus. One approach is to extract from the corpus generalized statistical knowledge that can be applied to new, unseen test sentences. A different approach is to simply memorize the bilingual corpus. This is called *translation memory* [1], and it provides excellent translation quality in the case of a “hit” (i.e., a test sentence to be translated has actually been observed before in the memorized corpus). However, it provides no output in the more frequent case of a “miss”.

While it is often unlikely that a test sentence will be found in a limited bilingual training corpus, it is much more likely that its *translation* will be found in a vast monolingual corpus. For example, note that the English sentence “She made quite a good job.” does not appear on Google’s Web index. But “She did quite a good job.” does appear. If both sentences are suggested to us as translations, we can therefore automatically prefer the latter simply because this string has been observed before in a large monolingual corpus of English text. Similarly, on the basis of the frequency of a phrase, we may prefer as translation of the French phrase “elle a beaucoup de cran” the English phrase “she has a lot of guts” as opposed to, say, “it has a lot of guts”, even if both phrases are found in a large monolingual corpus (Altavista’s Web index)¹. Their frequencies,

¹ “The road from Angkor Wat ... it has a lot of guts to call itself a road.”

however, differ by a factor of seven, and we can prefer the former translation on that basis. A similar idea has been proposed by Grefenstette [2] as an approach to lexical choice for machine translation. We take this idea a step further and propose the use of a vast monolingual corpus to validate full translations.

In this paper we introduce an alternative approach to machine translation that operationalizes the intuitions above. Our method uses a statistical machine translation system to produce alternative translations and a large monolingual corpus to (re)rank these alternative translations. We contribute to the field of machine translation in two ways:

- We introduce algorithms capable of counting the number of occurrences of a large number of translations (about 10^{300}) in a large monolingual sequence/corpus (about 1 billion words of English text).
- We show empirically that the accuracy of a statistical MT system can be improved if translated phrases/sentences are re-ranked according to their likelihood of occurring in a large monolingual corpus.

The algorithms we introduce can be also used to automatically construct parallel corpora starting from a translation table and a large monolingual corpus.

2 IBM Model 4

In this paper we use the IBM Model 4 [3]. For our purposes, the important feature of this model is that, for any given input French sentence, we can compute a large list of potential English translations (of order 10^{300} or even larger, see Section 3.2). IBM Model 4 revolves around the notion of word alignment over a pair of sentences (see Figure 1). A word alignment assigns a single English string position to each French word.

The word alignment in Figure 1 is shorthand for a hypothetical stochastic process by which an English string gets converted into a French string. There are several steps to be made. First, every English word is assigned a fertility. We delete from the string any word with fertility zero, we duplicate any word with fertility two, etc. Second, after each English word in the new string, we may insert an invisible NULL element with probability p_1 (typically about 0.02). The NULL element ultimately produces “spurious” French words. Next, we perform a word-for-word replacement of English words (including NULL) by French words, according to certain *translation probabilities* $t(f_j \mid e_i)$ (which together form a *translation table*, or T-table). Finally, the French words are permuted according to certain *distortion probabilities*.

3 Multiple String Matching against Large Sequences

From a computer science perspective, the problem we are trying to solve is simple: we are interested in determining the number of occurrences of a set of strings/translations $\{t_1, t_2, \dots, t_n\}$ in a large sequence/corpus S . When n and S are small, this is a trivial problem, and tools such as *grep* can easily solve the problem. Unfortunately, for large n , the problem becomes extremely challenging.

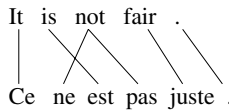


Fig. 1. Sample word alignment.

3.1 Naive Approaches

Simple *grep*

We ignore for the moment that we need to search for n strings. Even if one tries to search for all the occurrences of *one* string t_i in a corpus of 1 billion-words, *grep* will take about 30 minutes. Unfortunately, we do not have to search for 1 string, but for about 10^{300} . Searching sequentially using *grep* is clearly infeasible.

Regular Expressions and *egrep*

Another idea is to represent all the possible strings/translations in a regular expression. Given that these translations share a lot of information, one can expect that the resulting regular expression will be much more compact than a sequential enumeration of all possible translations. We developed an algorithm to compactly represent all the possible translations as a regular expression. For a 6-word French sentence, the regular expression that subsumes all its possible translations into English takes roughly 4 Mbytes. Such huge expressions cannot be processed by *egrep*.

Querying the Web

Another attractive solution seems to be that of querying directly the Web. After all, the Web can provide us access to the ultimate monolingual corpus; and search engines like Google answer queries in only a few mili-seconds. Unfortunately, one cannot do searches of large regular expressions on the Web, and therefore even if one search takes 1 ms, 10^{300} searches take an infeasible amount of time.

3.2 Multiple String Matching Using FSAs

In order to solve the multiple string matching problem, we decided to expand on a solution proposed initially by Knight and Al-Onaizan [4], which proposes a finite state acceptor (FSA) to compactly represent possible English translations of a French sentence. If the IBM Model 4 is used as translation model, then such an FSA has to account for all the stochastic steps described in Section 2. Fortunately, one can build small acceptors (FSAs) and transducers (FSTs) that account for each of these steps separately, and then compose them together to obtain an acceptor which accounts for all of them.

Representing Multiple Translations as FSAs

In the framework of IBM Model 4 we start with an English string and perform several steps to probabilistically arrive at a French string. When translating/decoding, we need to perform all the steps described in Section 2 in reverse order to obtain the English strings that may have produced the French sentence.

Assume that we are interested in representing compactly all English translations of the French phrase “un bon choix”. Since French and English have different word orders, we first need to generate all possible permutations of the French words. An FSA that accomplishes this task is presented in Figure 2(b).

The mapping between French and English words is often ambiguous. When translating from French into English, we can translate “un” as “a”, “an”, or even as NULL. We can build an FST to take into account the multiple translation possibilities. Given that we actually build probabilistic transducers, the probabilities associated with these possibilities can be incorporated. The T-table (Section 2) can be used to build a simple transducer: it has only one state and has one transition for each entry in the T-table (a highly simplified version is shown in Figure 2(a)). Composing this FST with the previous FSA results in an FSA modeling both the different word order and the word translation ambiguity phenomena (Figure 2(c)).

The story gets more complicated as one has to add new transducers for the other steps discussed in Section 2. For example, our French phrase “un bon choix” can be translated as “good choice” in English. Our model accomplishes this by considering the word “un” to be the translation of a NULL English word. A simple two-state automaton is used to model the NULL word insertions (see Figure 2(d)).

Finally, fertility also needs to be modeled by an FSA. In Figure 1, for example, the English word “not” is mapped into both “ne” and “pas”. This can be simulated by using the fertility 2 of “not” to first multiply it (i.e., create “not not” on the English side), and then translating the first one as “ne” and the second one as “pas”. A simple FSA (not shown here) can be used to model word fertilities.

The step-wise composition of these automata is shown in Figure 2. (Only very few possible translations are illustrated, and the probabilities are not included in the picture for readability.) For a given French sentence f , the final result of these operations is a non-deterministic FSA with epsilon transitions, which we generically call FSA_f^0 . For a 6-word French sentence f such as “elle me a beaucoup appris .”, the finite state acceptor we automatically generate has 464 states, 42139 arcs, and takes 1,172 Kbytes. The total number of paths (without cycles) is 10^{328} . There are a number of advantages to this representation:

- FSA_f^0 enumerates all possible English translations of f (according to the translation model).
- FSA_f^0 reflects the goodness of each translation e_i as assessed by the statistical model used to generate it; as Knight and Al-Onaizan [4] have shown, the probability of a path e through the FSA corresponds to the IBM-style transition model probability $P(f|e)$ [3].

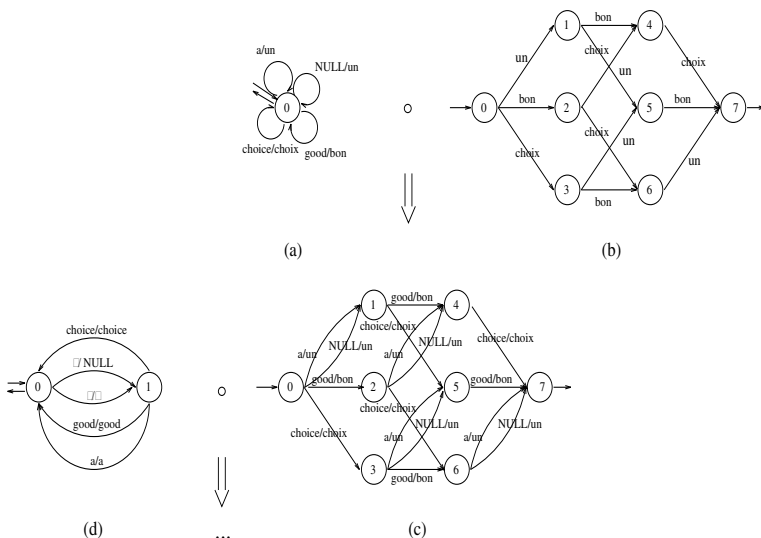


Fig. 2. Step-wise composition of FSAs/FSTs.

- FSA_f^0 can be used as a binary classifier for English strings/translations (“yes” if string e is a possible translation of f ; “no” otherwise).

A finite state machine built in this manner operates as a rudimentary statistical machine translation system. Given a French sentence f , it can output all its English translations e_i and their IBM-4 translation probabilities (modulo distortion probabilities).

Matching FSAs against Large Sequences

In the previous section, we have shown how to automatically build, for a given French sentence f , a finite state acceptor FSA_f^0 that encodes all possible English translations of f . The next step is to use FSA_f^0 to find all the occurrences of the possible English translations of f in a large monolingual corpus. In order to be able to perform the string matching operations, we need to modify the monolingual corpus such that all the English words unknown to FSA_f^0 are replaced by UNK in the monolingual corpus. The acceptor FSA_f^0 needs also to be slightly modified to account for the UNK token, and we call the resulted acceptor FSA_f^1 . We do not describe these modifications here due to lack of space.

A summary of all the operations is presented in Figure 3. From a French sentence f , using the parameters of a statistical translation model, a finite state acceptor FSA_f^0 is built. FSA_f^0 is further modified to yield FSA_f^1 . A large English corpus is taken sentence by sentence and modified such that all English words not known by FSA_f^0 are replaced by UNK. Each modified sentence is matched against FSA_f^1 , and for each sentence accepted by FSA_f^1 we store the string matched, and

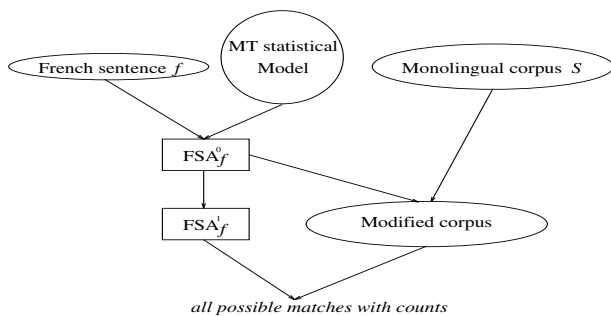


Fig. 3. Multiple string matching against a large corpus

also keep a count of each appearance. We end up with all possible translations of f that also occur in the corpus S , and their counts. The number of observed translations of f decreases from an order of magnitude of 10^{300} as proposed by the translation model to an order of magnitude of 10^3 - 10^6 .

All these operations can be performed using off-the-shelf software. We use a publicly available finite-state package² to perform the composition operations which yield FSA_f^0 . We use the same package for the string matching problem: FSA_f^1 is loaded into memory and matched sentence by sentence against the corpus, performing the acceptance test.

4 Time Performance Issues

The multi-string matching method described in Section 3 is capable of finding all translations of a given French sentence in a large monolingual, English corpus, but it needs to be performed in parallel in order to be usable. A parallel version of our algorithm, run on a Linux cluster of 192 nodes and 384 CPUs (Pentium 3, 766MHz & 866MHz), obtains linear speed-up in run-time with the increase in the number of processors. For a 6-word French sentence we obtain the following reductions in time as we increase the number of processors:

1 Processor:	30 hrs. (1800 min.)
10 Processors:	3 hrs. (180 min.)
100 Processors:	0.3 hrs. (18 min.)

The parallelization method is straightforward: given the total number of sentences in the corpus, one has to assign an equal number of sentences to each processor, run in parallel the algorithm on each such corpus, and then collect and sum up the results.

² <http://www.isi.edu/licensed-sw/carmel/index.html>

5 Performance Evaluation

In order to assess whether our translation method can improve the performance of IBM Model 4 we collected a corpus of 1.6 billion words of English by collecting various corpora made available by LDC³. The parameters of the statistical model were trained on 500,000 parallel sentences from the Hansard genre using GIZA⁴. We translated 101 in-domain French sentences (of length 6) taken from the Hansard genre, and 110 out-of-domain French sentences (of length 6) taken from Le Monde.

We translated these sentences using four different methods: two of them were previously published methods using only the parameters of the statistical model and different decoding algorithms – Stack and Greedy [5] – and were publicly available for downloading⁵. The other two methods were variations of the method using a Big monolingual Corpus (therefore called BC here): a pure BC method, and a semi-automated method BC⁺ where we looked at the first 10 translations proposed by the BC method for each sentence and selected by hand the best one.

5.1 The BC Method

For a given French sentence suppose there is a possible translation occurring in the monolingual corpus. For each such translation we store:

- The translation model probability which ignores distortions (TM^-)
- The number of occurrences in the monolingual corpus (NO)
- The alignment for each translation

The formula $NO \times TM^-$ is used for a pre-ranking from which the first 500 candidates are extracted. On each of these candidates we compute a language model probability (LM) using trigrams and also (using the alignment) a translation model probability which includes distortions (TM).

The last step is to re-rank the top 500 candidates using a formula $F(NO, LM, TM)$. We currently re-rank according to the formula $NO \times LM \times TM$. The formula F can be trained to yield optimal results, which we leave for future work.

5.2 Evaluation

Table 1 shows the number of perfect translations obtained on both corpora by each of the methods used. Here, “perfect” refers to a human-judged translation that transmits all of the meaning of the source sentence using flawless target-language syntax. The table also shows the number of untranslated/unscored sentences for each corpus.

³ <http://www ldc.upenn.edu/>

⁴ <http://www.clsp.jhu.edu/ws99/projects/mt/>

⁵ <http://www.isi.edu/natural-language/projects/rewrite/>

Table 1. Translation scores obtained by different methods.

	In-domain		Out-of-domain	
	perfect	unsc.	perfect	unsc.
Stack	40	0	16	0
Greedy	53	0	18	0
BC	43	7	24	33
BC ⁺	62	7	32	33

Table 2. Confusion matrices for the BC and Greedy methods.

	In-domain		Out-of-domain	
	G perf.	G err.	G perf.	G err.
BC perf.	31	12	10	14
BC err.	22	29	8	45

The BC method, which we introduced in this paper, produces 43 perfect translations out of the 101 sentences of the in-domain corpus and returns zero translations for 7, which means 42.5% recall and 45.7% precision. Returning zero translations is a failure of our method which is discussed in Section 6. The performance of the Greedy method is higher for in-domain sentences (52.4% recall and precision).

For the 110 out-of-domain sentences, the BC method produces 24 perfect translations and returns zero translations for 33, which means 21.8% recall and 31.1% precision. The Greedy method has a performance of only 16.3% recall and precision on this corpus. These results show that, although the Greedy method performs well on in-domain sentences, it is more dependent than the BC method on the genre on which the parameters are trained. The BC method depends less on the statistical parameters and therefore has a better performance for out-of-domain sentences.

The BC⁺ method outperforms the Stack and the Greedy methods for both in-domain and out-of-domain sentences. It has 61.3% recall and 65.9% precision for in-domain sentences, and 29.1% recall and 41.5% precision for out-of-domain sentences. This proves that, although the formula used for re-ranking is perhaps not optimal, the method proposed here can significantly improve translation accuracy.

Another useful comparison is shown in Table 2. It indicates the amount of overlap for the perfect translations produced by the BC and Greedy methods. The confusion matrices prove that these two methods yield quite orthogonal results, i.e., their results are produced independently and the correct translations do not necessarily overlap. For example, the BC method produces 12 in-domain and 14 out-of-domain perfect translations. These translations are not found by the Greedy decoder because they have a lower probability according to the translation model.

6 Discussion

In this section we examine a common cause for failure in our system, and also briefly discuss the possible use of this translation method for other language pairs.

A major cause of failure for the BC method is the gaps in the corpus S . By the very idea of this method – trying to find phrases in S that are translations of the initial sentence – the algorithm can fail to find any such possible translation, returning zero proposed translations. This type of failure has several possible fixes. One is to keep increasing the size of the corpus S beyond 1 billion words of magnitude. Intuitively this gives our algorithm an increased chance of finding good translation proposals. Another possible fix is to incorporate the BC method, together with other translation methods, into a multi-engine system which combines the strengths of each individual method.

And yet another possible approach to fixing this type of failure is to find a reliable mechanism for splitting up sentences into “independent” sub-parts (such as clauses, or elementary textual units [6]), and then translate them individually. We suspect that such an approach would also allow for the method proposed here to scale up to longer sentences without losing much in the translation accuracy.

The method presented here has the potential to work for other language pairs as well, as long as a large monolingual corpus of the target language and a translation table for the language pair are available. The only extra condition that seems to be required is that the monolingual corpus is comparable content-wise with the corpus from which the sentences to be translated are extracted.

7 Building New Parallel Corpora

Parallel corpora are expensive resources that are time-consuming to build by humans; yet, they are crucial for building high-performance statistical machine translation systems. Although as a translation mechanism our method is both time and resource consuming, we believe it can also be used to automatically construct parallel corpora quicker and cheaper. We hypothesize that new phrase/sentence pairs aligned by our method can be extracted and used for training, in order to improve the estimates of the parameters of a statistical model.

References

1. Sprung, R., ed.: Translating Into Success: Cutting-Edge Strategies For Going Multilingual In A Global Age. John Benjamins Publishers (2000) 155
2. Grefenstette, G.: The world wide web as a resource for example-based machine translation tasks. In: ASLIB, Translating and the Computer 21, London (1999) 156
3. Brown, P., Della Pietra, S., Della Pietra, V., Mercer, R.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19** (1993) 263–311 156, 158

4. Knight, K., Al-Onaizan, Y.: Translation with finite-state devices. In: Proceedings of the 4th AMTA Conference. (1998) 157, 158
5. Germann, U., Jahr, M., Knight, K., Marcu, D., Yamada, K.: Fast decoding and optimal decoding for machine translation. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01), Toulouse, France (2001) 161
6. Marcu, D.: A surface-based approach to identifying discourse markers and elementary textual units in unrestricted texts. In: Proceedings of the COLING/ACL-98 Workshop on Discourse Relations and Discourse Markers, Montreal, Canada (1998) 163

Semi-automatic Compilation of Bilingual Lexicon Entries from Cross-Lingually Relevant News Articles on WWW News Sites

Takehito Utsuro, Takashi Horiuchi,
Yasunobu Chiba, and Takeshi Hamamoto

Department of Information and Computer Sciences,
Toyohashi University of Technology
Tenpaku-cho, Toyohashi 441-8580, Japan
{utsuro,takashi,chiba,hamamo}@cl.ics.tut.ac.jp

Abstract. For the purpose of overcoming resource scarcity bottleneck in corpus-based translation knowledge acquisition research, this paper takes an approach of semi-automatically acquiring domain specific translation knowledge from the collection of bilingual news articles on WWW news sites. This paper presents results of applying standard co-occurrence frequency based techniques of estimating bilingual term correspondences from parallel corpora to relevant article pairs automatically collected from WWW news sites. The experimental evaluation results are very encouraging and it is proved that many useful bilingual term correspondences can be efficiently discovered with little human intervention from relevant article pairs on WWW news sites.

1 Introduction

Translation knowledge acquisition from parallel/comparative corpora [4] is one of the most important research topics of corpus-based MT. This is because it is necessary for an MT system to (semi-)automatically increase its translation knowledge in order for it to be used in the real world situation. One limitation of the corpus-based translation knowledge acquisition approach is that the techniques of translation knowledge acquisition heavily rely on availability of parallel/comparative corpora. However, the sizes as well as the domain of existing parallel/comparative corpora are limited, while it is very expensive to manually collect parallel/comparative corpora. Therefore, it is quite important to overcome this resource scarcity bottleneck in corpus-based translation knowledge acquisition research.

In order to solve this problem, this paper focuses on bilingual news articles on WWW news sites as a source for translation knowledge acquisition. In the case of WWW news sites in Japan, Japanese as well as English news articles are updated everyday. Although most of those bilingual news articles are not parallel even if they are from the same site, certain portion of those bilingual news articles share

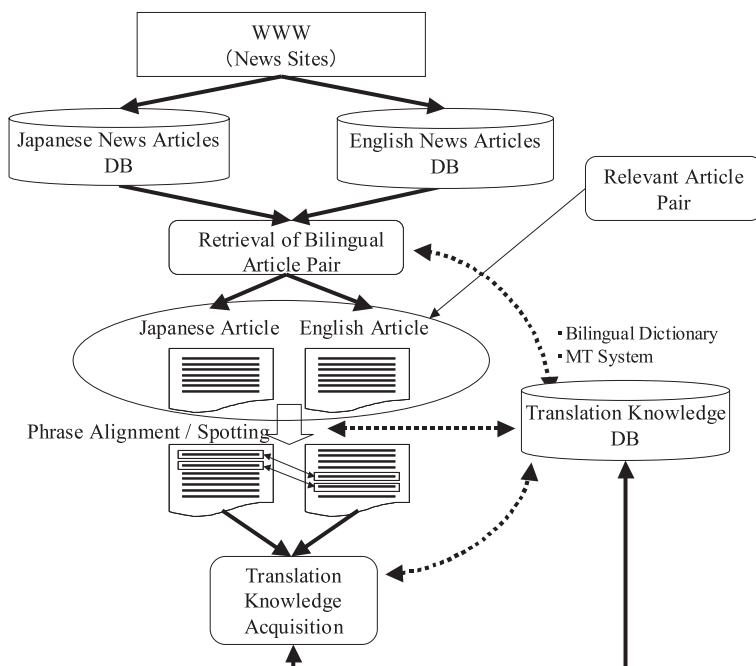


Fig. 1. Translation Knowledge Acquisition from WWW News Sites: Overview

their contents or at least report quite relevant topics. Based on this observation, we take an approach of semi-automatically acquiring translation knowledge of domain specific named entities, event expressions, and collocational functional expressions from the collection of bilingual news articles on WWW news sites.

Figure 1 illustrates the overview of our framework of translation knowledge acquisition from WWW news sites. First, pairs of Japanese and English news articles which report identical contents or at least closely related contents are retrieved. (Hereafter, we call pairs of bilingual news articles which report identical contents as “*identical*” pair, and those which report closely related contents (e.g., a pair of a crime report and the arrest of its suspect) as “*relevant*” pair.) Then, by applying term/phrase alignment techniques to Japanese and English news articles, various kinds of translation knowledge are acquired. In the process of translation knowledge acquisition, we allow human intervention if necessary. Especially, we aim at developing user interface facilities for efficient semi-automatic acquisition of translation knowledge, where previously studied techniques of translation knowledge acquisition from parallel/comparative corpora [4] are integrated in an optimal fashion.

Within this framework of translation knowledge acquisition from WWW news sites, this paper studies issues regarding cross-language retrieval and collection of “*identical*”/ “*relevant*” article pairs. We also present results of apply-

ing standard co-occurrence frequency based techniques of estimating bilingual term correspondences from parallel corpora [4] to those automatically collected “*identical*”/ “*relevant*” article pairs. The experimental evaluation results are very encouraging and it is proved that many useful bilingual term correspondences can be efficiently discovered with little human intervention from relevant article pairs on WWW news sites. Details of those evaluation results are presented.

2 Cross-Language Retrieval of Relevant News Articles

This section gives the overview of our framework of cross-language retrieval of relevant news articles from WWW news sites. First, from WWW news sites, both Japanese and English news articles within certain range of dates are retrieved. Let d_J and d_E denote one of the retrieved Japanese and English articles, respectively. Then, each English article d_E is translated into a Japanese document d_J^{MT} by some commercial MT software¹. Each Japanese article d_J as well as each Japanese translation d_J^{MT} of the English articles are next segmented into word sequences by the Japanese morphological analyzer CHASEN (<http://chasen.aist-nara.ac.jp/>), and word frequency vectors v_J and v_J^{MT} are generated². Then, cosine similarities between v_J and v_J^{MT} are calculated³ and pairs of articles d_J and d_E (d_J^{MT}) which satisfy certain criterion are considered as candidates for “*identical*” or “*relevant*” article pairs.

3 Acquisition of Bilingual Term Correspondences from Relevant News Articles

3.1 Estimating Bilingual Term Correspondences

This section briefly describes the method of estimating bilingual term correspondences from the results of retrieving cross-lingually relevant English and Japanese news articles. As will be described in section 4.1, on WWW news sites in Japan, the number of articles updated per day is far greater (5~30 times) in Japanese than in English. Thus, it is much easier to find cross-lingually relevant articles for each *English* query article than for each *Japanese* query article. Considering this fact, we estimate bilingual term correspondences from the results of cross-lingually retrieving relevant *Japanese* articles with *English* query articles.

¹ As the commercial MT software, we chose English-Japanese Japanese-English Translation Software HONYAKUDAMASHII for Linux/BSD, OMRON SOFTWARE Co., Ltd, which is the one with slightly better performance than the others.

² After removing the most frequent 26 *hiragana* functional expressions as stop words, word frequency vectors are generated only from nouns and verbs.

³ It is also quite possible to translate Japanese news articles into English and to calculate similarities of word frequency vectors in English side.

For an English query article d_E^i , let D_J^i denote the set of Japanese articles with cosine similarities higher than or equal to a certain lower bound L_d :

$$D_J^i = \left\{ d_J \mid \cos(d_E^i, d_J) \geq L_d \right\}$$

Then, we concatenate constituent Japanese articles of D_J^i into one article $D_J'^i$, and construct a pseudo-parallel corpus PPC_{EJ} of English and Japanese articles:

$$PPC_{EJ} = \left\{ \langle d_E^i, D_J'^i \rangle \mid D_J^i \neq \emptyset \right\}$$

Next, we apply standard techniques of estimating bilingual term correspondences from parallel corpora [4] to this pseudo-parallel corpus PPC_{EJ} . First, we extract monolingual (possibly compound) terms t_E and t_J which satisfy requirements on frequency lower bound and the upper bound of the number of constituent words. Then, based on the contingency table of co-occurrence frequencies of t_E and t_J below, we estimate bilingual term correspondences according to the statistical measures such as the mutual information, the ϕ^2 statistic, the dice coefficient, and the log-likelihood ratio [4].

	t_J	$\neg t_J$
t_E	$freq(t_E, t_J) = a$	$freq(t_E, \neg t_J) = b$
$\neg t_E$	$freq(\neg t_E, t_J) = c$	$freq(\neg t_E, \neg t_J) = d$

We compare the performance of those four measures, where the ϕ^2 statistic and the log-likelihood ratio perform best, the dice coefficient the second best, and the mutual information the worst. In section 4.3, we show results with the ϕ^2 statistic:

$$\phi^2(t_E, t_J) = \frac{(ad - bc)^2}{(a + b)(a + c)(b + d)(c + d)}$$

3.2 Semi-automatic Acquisition of Bilingual Term Correspondences

This section describes the method of semi-automatic acquisition of bilingual term correspondences from the results of estimating bilingual term correspondences. Since our source of compiling bilingual lexicon entries is not clean parallel corpus, but artificially generated noisy pseudo-parallel corpus, it is difficult to compile bilingual lexicon entries full-automatically. In order to reduce the amount of human intervention necessary for selecting correctly estimated bilingual term correspondences, we divide the whole set of estimated bilingual term correspondences into subsets according to the following two criteria. First, we divide the whole set of estimated bilingual term correspondences into subsets, where each subset consists of English and Japanese term pairs which have a common English term. Next, we define the relation $t \succeq t'$ between two terms t and t' as t being identical with t' or the term t' constituting a part of the compound term

t . Then, for each English term t_E , only when any other English term t'_E does not satisfy the relation $t'_E \succeq t_E$, we construct the set $TP(t_E)$ of English and Japanese term pairs which have t_E or its sub-sequence term in the English side and satisfy the requirements on (co-occurrence) frequencies and term length in their constituent words as below:

$$TP(t_E) = \left\{ \langle t'_E, t_J \rangle \mid t_E \succeq t'_E, freq(t_E) \geq L_f^E, freq(t_J) \geq L_f^J, \right. \\ \left. freq(t_E, t_J) \geq L_f^{EJ}, length(t_E) \leq U_l^E, length(t_J) \leq U_l^J \right\}$$

We call the shared English term t_E of the set $TP(t_E)$ as *index*.

Next, all the sets $TP(t_E^1), \dots, TP(t_E^m)$ are sorted in descending order of the maximum value $\hat{\phi}^2(TP(t_E))$ of ϕ^2 statistic of their constituent term pairs:

$$\hat{\phi}^2(TP(t_E)) = \max_{\langle t_E, t_J \rangle \in TP(t_E)} \phi^2(t_E, t_J)$$

Then, each set $TP(t_E^i)$ is examined by hand according to whether or not it includes correct bilingual term correspondences. Finally, we evaluate the following rate of containing correct bilingual term correspondences:

$$\begin{array}{l} \text{rate of} \\ \text{containing} \\ \text{correct} \\ \text{bilingual term} \\ \text{correspondences} \end{array} = \frac{\left| \left\{ TP(t_E) \mid \text{correct bilingual term correspondence} \right. \right. \\ \left. \left. \langle t_E, t_J \rangle \in TP(t_E) \right\} \right|}{\left| \left\{ TP(t_E) \mid TP(t_E) \neq \emptyset \right\} \right|} \quad (1)$$

3.3 Example

Figure 2 illustrates the underlying idea of semi-automatic selection of correct bilingual term correspondences, with the help of browsing cross-lingually relevant article pairs. Suppose that an English compound term “Tokyo District Court” is chosen as the *index term* t_E . The figure lists the term pairs t_E and t_J with high values of ϕ^2 statistic in descending order, together with $freq(t_E)$, $freq(t_J)$, $freq(t_E, t_J)$, and $\phi^2(t_E, t_J)$. In this case, t_J with the highest value of ϕ^2 statistic is the correct Japanese translation of “Tokyo District Court”. A human operator can select an arbitrary pair of English and Japanese terms t_E and t_J and then browse an English and Japanese article pair d_E and d_J , each of which contains t_E and t_J , respectively, and satisfies the similarity requirement $\cos(d_E, d_J) \geq L_d$. When the human operator browses such an article pair d_E and d_J , titles of English articles which contain t_E are first listed, and then, for each of the English articles, titles of Japanese articles which contain t_J and satisfy the similarity requirement are listed. Browsing through the title list as well as the body texts of the English and Japanese article pairs, the human operator can easily judge whether the selected term pair t_E and t_J is actually correct translation of each other. Even when the selected term pair is not correct translation, it is usually quite easy for the human operator to discover true

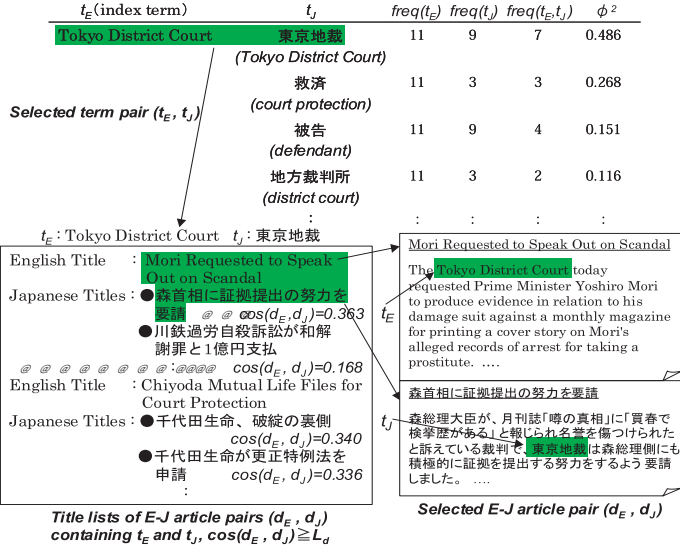


Fig. 2. Example of Semi-Automatic Selection of Bilingual Term Correspondences with Browsing Cross-Linguely Relevant Article Pairs

term correspondence if the selected article pair reports closely related contents. Otherwise, the human operator can quickly switch the article pair to the one which reports closely related contents.

4 Experimental Evaluation

4.1 Japanese-English Relevant News Articles on WWW News Sites

Table 1. Total # of Days, Total/Average # of Articles / Average Article Size / # of Reference Article Pairs for CLIR Evaluation

Site	Total # of Days		Total # of Articles		Average # of Articles per Day		Average Article Size (bytes)		# of Reference Article Pairs for CLIR Evaluation	
	Eng	Jap	Eng	Jap	Eng	Jap	Eng	Jap	Identical	Relevant
A	562	578	607	21349	1.1	36.9	1087.3	759.9	24	33
B	162	168	2910	14854	18.0	88.4	3135.5	836.4	28	82
C	162	166	3435	16166	21.2	97.4	3228.9	837.7	28	31

We collected Japanese and English news articles from three WWW news sites A, B, and C. Table 1 shows the total number of collected articles and the range of dates of those articles represented as the number of days. Table 1 also shows the number of articles updated in one day, and the average article size. The number of Japanese articles updated in one day are far greater (5~30 times)

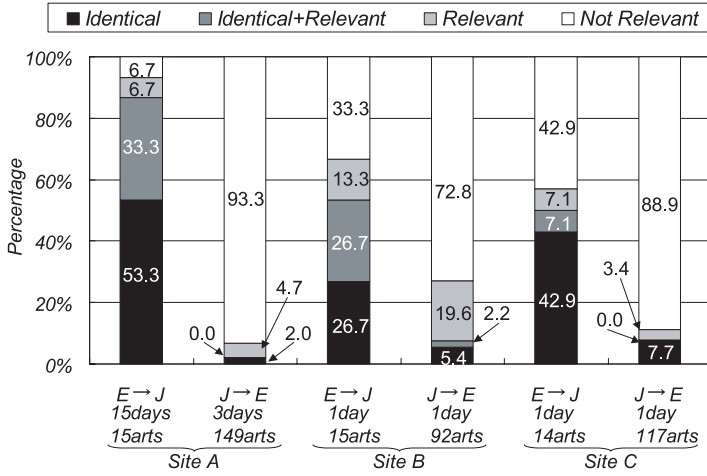


Fig. 3. Availability of Cross-Lingually “Identical”/“Relevant” Articles

than that of English articles. In addition to that, the table gives the numbers of reference “identical”/“relevant” article pairs manually collected for the evaluation of cross-language retrieval of relevant news articles. This evaluation result will be presented in the next section. In the case of those reference article pairs, the difference of dates between “identical” article pairs is less than ± 5 days, and that between “relevant” article pairs is around ± 10 days.

Next, Figure 3 shows rates of whether cross-lingually “identical” or “relevant” articles are available or not for each retrieval query article, where the following counts are recorded and their distributions are shown in the figure: i) the number of queries for which at least one “identical” article is available, but not any “relevant” article, ii) the number of queries for which at least one “identical” article and one “relevant” article are available, iii) the number of queries for which at least one “relevant” article is available, but not any “identical” article, iv) the number of queries for which neither “identical” nor “relevant” article is available. As can be clearly seen from these results, since the number of Japanese articles are far greater than that of English articles, the availability rate in Japanese-to-English retrieval is much lower than that in English-to-Japanese retrieval. The availability rate (either “identical” or “relevant”) in Japanese-to-English retrieval is around 10~30%, while in English-to-Japanese retrieval, that for “identical” articles is more than 50%, and that for either “identical” or “relevant” increases by around 10% and more. These results guarantee that cross-lingually “identical” news articles are available in the direction of English-to-Japanese retrieval for more than half of the retrieval query English articles.

4.2 Cross-Language Retrieval of Relevant News Articles

Next, we evaluate the performance of cross-language retrieval of “identical” / “relevant” reference article pairs given in Table 1. In the direction of English to Japanese cross-language retrieval, precision/recall rates of the reference

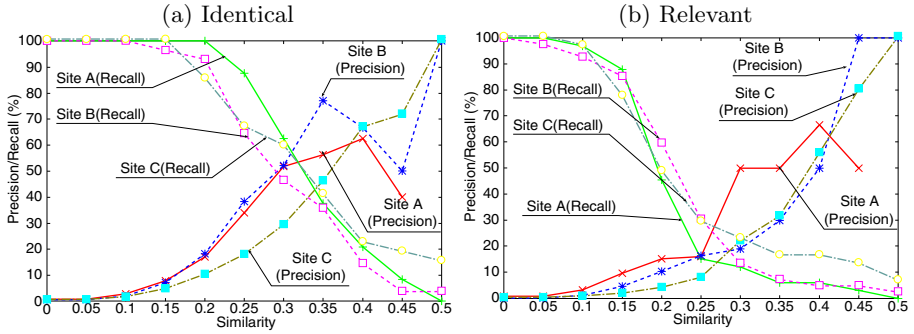


Fig. 4. Precision/Recall of Cross-Language Retrieval of Relevant News Articles (Article Similarity $\geq L_d$)

“*identical*”/ “*relevant*” articles against those with the similarity values above the lower bound L_d are measured, and their curves against the changes of L_d are shown in Figure 4. The difference of dates of English and Japanese articles is given as the maximum range of dates, with which all the cross-lingually “*identical*”/ “*relevant*” articles can be discovered (less than ± 5 days for the “*identical*” article pairs and around ± 10 days for the “*relevant*” article pairs). Let DP_{ref} denote the set of reference article pairs within the range of dates, the precise definitions of the precision and recall rates of this task are given below:

$$\text{precision} = \frac{|\{d_J \mid \exists d_E, \langle d_E, d_J \rangle \in DP_{ref}, \cos(d_E, d_J) \geq L_d\}|}{|\{d_J \mid \exists d_E \exists d'_J, \langle d_E, d'_J \rangle \in DP_{ref}, \cos(d_E, d_J) \geq L_d\}|}$$

$$\text{recall} = \frac{|\{d_J \mid \exists d_E, \langle d_E, d_J \rangle \in DP_{ref}, \cos(d_E, d_J) \geq L_d\}|}{|\{d_J \mid \exists d_E, \langle d_E, d_J \rangle \in DP_{ref}\}|}$$

In the case of “*identical*” article pairs, Japanese articles with the similarity values above 0.4 have precision of around 40% or more⁴.

4.3 Semi-automatic Acquisition of Bilingual Term Correspondences from Relevant News Articles

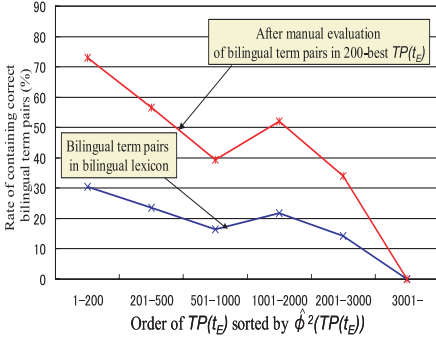
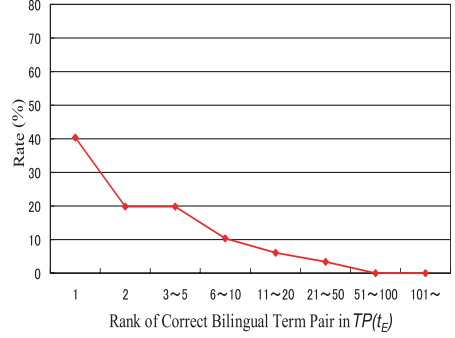
In this section, we evaluate our framework of semi-automatic acquisition of bilingual term correspondences from relevant news articles. For the news sites A, B, and C, and for several lower bounds L_d of the similarity between English and Japanese articles, Table 2 shows the numbers of English and Japanese articles which satisfy the similarity lower bound⁵. Then, under the conditions

⁴ We are now working on examining usefulness of additional clues such as titles, pronunciation of foreign names, and numerical expressions, and furthermore on incorporating them for the purpose of improving the performance of cross-language retrieval.

⁵ It can happen that one Japanese article is retrieved by more than one English query articles. In such cases, the occurrence of Japanese articles is duplicated.

Table 2. Numbers of Japanese/English Articles Pairs with Similarity Values above the Lower Bounds

Site	A				B		C	
Lower Bound L_d of Articles' Sim	0.25	0.3	0.4	0.5	0.4	0.5	0.4	0.5
Difference of Dates (days)	± 4				± 3		± 2	
# of English Articles	473	362	190	74	415	92	453	144
# of Japanese Articles	1990	1128	377	101	631	127	725	185

(a) Rates of Containing Correct Bilingual Term Pairs (Site A, $L_d = 0.3$)(b) Ranks of 146 Correct Bilingual Term Pairs within 200-best $TP(t_E)$, Sorted by ϕ^2 (Site A, $L_d = 0.3$)**Fig. 5.** Evaluation Results using Bilingual Term Pairs in a Bilingual Lexicon / by Manual Evaluation

$L_f^E = L_f^J = 3, L_f^{EJ} = 2, U_l^E = U_l^J = 5$ (the difference of dates of English and Japanese articles is given as the maximum range of dates, with which all the cross-lingually “*identical*” articles can be discovered), the sets $TP(t_E)$ are constructed and the “rate of containing correct bilingual term correspondences” in the equation (1) (section 3.2) is evaluated.

For the site A with the similarity lower bound $L_d = 0.3$, the rates of containing correct bilingual term pairs taken from an existing bilingual lexicon (Eijiro Ver.37, 850,000 entries, <http://member.nifty.ne.jp/eijiro/>) are shown in Figure 5 (a) as “Bilingual term pairs in bilingual lexicon”. This result supports the usefulness of ϕ^2 statistic in this task, since the rate of containing correct bilingual term pairs tends to decrease as the order of $TP(t_E)$ sorted by $\hat{\phi}^2(TP(t_E))$ becomes lower. Furthermore, topmost 200 $TP(t_E)$ according to the ϕ^2 statistic $\hat{\phi}^2(TP(t_E))$ are examined by hand and 146 bilingual term pairs contained in the topmost 200 $TP(t_E)$ are judged as correct. This manual evaluation result indicates that, compared with the bilingual term pairs found in the existing bilingual lexicon, about 1.4 times those found in the existing bilingual lexicon can be acquired from the topmost 200 $TP(t_E)$. Figure 5 (a) also shows the estimated plot of “After manual evaluation of bilingual term pairs in 200-best $TP(t_E)$ ”, which is the rate of containing correct bilingual term pairs taken from the existing bilingual lexicon, multiplied by the ratio of about 2.4 (i.e., 146 of

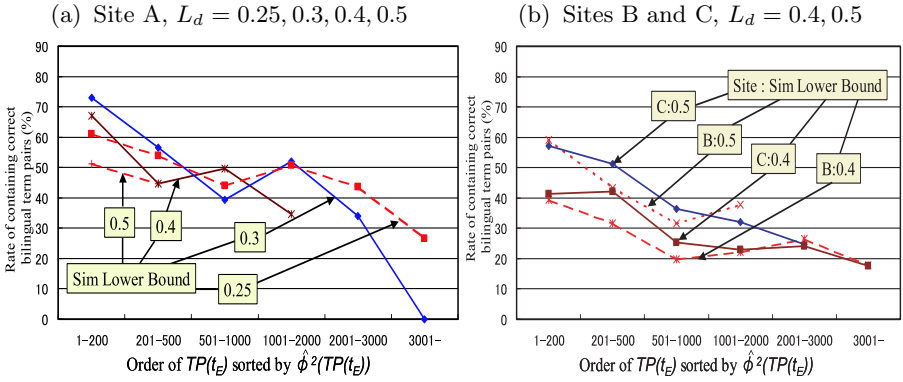


Fig. 6. Rates of Containing Correct Bilingual Term Pairs

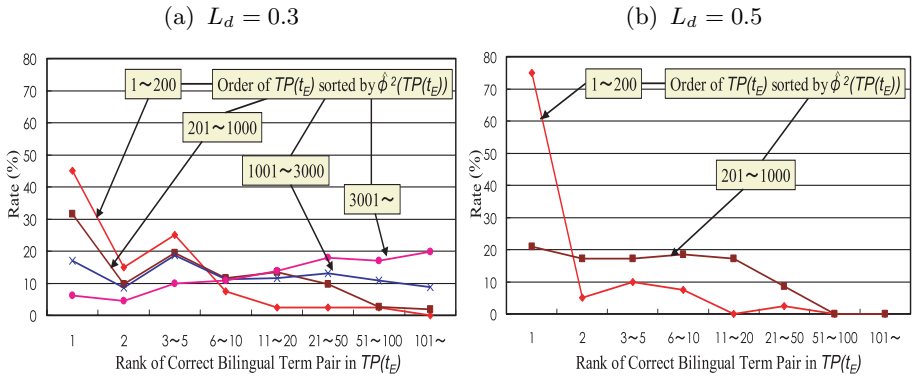


Fig. 7. Ranks of Correct Bilingual Term Pairs within a $TP(t_E)$, Sorted by ϕ^2 (Site A, Bilingual Term Pairs taken from a Bilingual Lexicon)

those judged as correct by manual evaluation/61 of those found in the existing bilingual lexicon).

Next, for the similarity lower bound $L_d = 0.25, 0.3, 0.4, 0.4$ (site A) and $L_d = 0.4, 0.5$ (sites B and C), estimated plots of rates of containing correct bilingual term pairs judged as correct by manual evaluation (i.e., those for correct bilingual term pairs taken from the existing bilingual lexicon, multiplied by the ratio of about 2.4) are shown in Figure 6. As can be seen from these results, the lower the similarity lower bound L_d is, the more the number of articles retrieved is and the more the number of candidate bilingual term pairs is, which is indicated by the difference of the lengths of those plots. For the site A, and for the sites B and C when the similarity lower bound $L_d = 0.5$, rates of containing correct bilingual term pairs are over 40% within the top 500 $TP(t_E)$. These rates are high enough for efficient human intervention in semi-automatic compilation of bilingual lexicon entries. Furthermore, the rates of containing correct bilingual term pairs are comparable among those three sites, even though the availability rates of cross-lingually “identical”/“relevant” articles are much lower for the sites

B and C than for the site A (Figure 3). This result is very encouraging because news sites with less availability rates of cross-lingually “*identical*”/“*relevant*” articles are still very useful in our framework and it proves the effectiveness of our approach⁶.

Finally, we evaluate the rank of correct bilingual term correspondences within each set $TP(t_E)$, sorted by ϕ^2 statistic. Within a set $TP(t_E)$, estimated Japanese term translation t_J are sorted by $\phi^2(t_E, t_J)$, and the ranks of correct Japanese translation of t_E are recorded. For the site A with the similarity lower bound $L_d = 0.3$, Figure 5 (b) shows the distribution of the ranks of correctly estimated Japanese terms for the 146 bilingual term pairs, which are contained in the topmost 200 $TP(t_E)$ and judged as correct. This result indicates that about 90% of those correct bilingual term pairs are included within the 10-best candidates in each $TP(t_E)$. For the site A with the similarity lower bound $L_d = 0.3, 0.5$, Figure 7 also shows this distribution for the correct bilingual term pairs taken from the existing bilingual lexicon. These results also support the usefulness of ϕ^2 statistic in this task, since the relative orders of correct bilingual term pairs tend to become lower as the order of $TP(t_E)$ sorted by $\hat{\phi}^2(TP(t_E))$ becomes lower. The criterion of the ϕ^2 statistic can be regarded as quite effective in reducing the amount of human intervention necessary for selecting correctly estimated bilingual term correspondences⁷. Furthermore, comparing the results of Figure 7 (a) and (b), the relative orders of correct bilingual term pairs become significantly higher when the similarity lower bound L_d is high. This result claims that the efficiency of semi-automatic acquisition of bilingual term pairs greatly depends on the accuracy of retrieving cross-lingually relevant news articles.

5 Related Works

Previously studied techniques of estimating bilingual word correspondences from non-parallel corpora (e.g., [1]) are based on the idea that semantically similar words appear in similar contexts. In those techniques, frequency information of contextual words co-occurring in the monolingual text is stored and their similarity is measured across languages. One of the most important difference between our approach and those techniques for translation knowledge acquisition from non-parallel corpora is that, we estimate bilingual term correspondences after selecting relevant article pairs, while in the latter techniques, co-occurrence frequency information is collected from the whole monolingual text. One of the

⁶ We also evaluate Japanese used as the language of the *index* term of each set TP and compare the “rate of containing correct bilingual term correspondences” with those with English *index* terms. Since the number of Japanese articles is far greater than that of English articles, this rate with Japanese *index* terms becomes lower for the similarity lower bounds $L_d \leq 0.4$.

⁷ It is also very important to note that the results of this paper can be easily improved by employing more sophisticated techniques of estimating bilingual compound term correspondences from parallel corpora (e.g., [2]), especially in the performance of selecting appropriate monolingual compound terms in each language.

major contribution of our work to the community of translation knowledge acquisition from parallel/comparable corpora is that: we showed even with standard techniques of estimating bilingual term correspondences from *parallel* corpora, many useful bilingual term correspondences can be efficiently discovered with little human intervention from relevant article pairs on WWW news sites. Furthermore, our results can be improved by incorporating those techniques based on co-occurrence frequency information in monolingual text (e.g., [1]), which are robust against noisy parallel corpora like those used in our work.

Related works on automatic document alignment between two languages include [3], which, in the context of cross-language information retrieval (CLIR) research, proposed to apply bootstrapping technique to an existing corpus-based CLIR approach for the task of extracting bilingual text pairs. Previous works on automatic document alignment mainly focused on the performance of automatic document alignment. Another type of related works include an approach of collecting partially bilingual texts from WWW [5]. One advantage of this approach is that it is applicable to various domains that infrequently become topics of news articles, although there might exist the case that the quality of translation by non-natives is possibly low. On the other hand, one of the advantages of our approach of employing bilingual news articles on WWW news sites as a source for translation knowledge acquisition is that high translation quality is guaranteed and articles of up-to-date topics are updated everyday.

6 Conclusion

Within the framework of translation knowledge acquisition from WWW news sites, this paper presented results of applying standard co-occurrence frequency based techniques of estimating bilingual term correspondences from parallel corpora to relevant article pairs automatically collected from WWW news sites. The experimental evaluation results were very encouraging and it was proved that many useful bilingual term correspondences can be efficiently discovered with little human intervention from relevant article pairs on WWW news sites.

References

1. Fung, P. and Yee, L. Y.: An IR Approach for Translating New Words from Nonparallel, Comparable Texts, *Proc. 17th COLING and 36th ACL* (1998) 414–420 175, 176
2. Haruno, M., Ikehara, S. and Yamazaki, T.: Learning Bilingual Collocations by Word-Level Sorting, *Proc. 16th COLING* (1996) 525–530 175
3. Masuichi, H., Flournoy, R., Kaufmann, S. and Peters, S.: A Bootstrapping Method for Extracting Bilingual Text Pairs, *Proc. 18th COLING* (2000) 1066–1070 176
4. Matsumoto, Y. and Utsuro, T.: Lexical Knowledge Acquisition, Dale, R., Moisl, H. and Somers, H. (eds.), *Handbook of Natural Language Processing*, chapter 24, Marcel Dekker Inc. (2000) 563–610 165, 166, 167, 168, 168
5. Nagata, M., Saito, T. and Suzuki, K.: Using the Web as a Bilingual Dictionary, *Proc. Workshop on Data-driven Methods in Machine Translation* (2001) 95–102 176

Bootstrapping the Lexicon Building Process for Machine Translation between ‘New’ Languages

Ruvan Weerasinghe*

Department of Computer Science,
University of Colombo, Sri Lanka
{arw@mail.cmb.ac.lk}

Abstract. The cumulative effort over the past few decades that have gone into developing linguistic resources for tasks ranging from machine readable dictionaries to translation systems is enormous. Such effort is prohibitively expensive for languages outside the (largely) European family. The possibility of building such resources automatically by accessing electronic corpora of such languages are therefore of great interest to those involved in studying these ‘new’ - ‘lesser known’ languages. The main stumbling block to applying these data driven techniques directly is that most of them require large corpora rarely available for such ‘new’ languages. This paper describes an attempt at setting up a bootstrapping agenda to exploit the scarce corpus resources that may be available at the outset to a researcher concerned with such languages. In particular it reports on results of an experiment to use state-of-the-art data-driven techniques for building linguistic resources for Sinhala – a non-European language with virtually no electronic resources.

1 Introduction

Machine processing of Natural (Human) Languages has a long tradition, benefiting from decades of manual and semi-automatic analysis by linguists, sociologists, psychologists and computer scientists among others. This cumulative effort has seen fruit in recent years in the form of publicly available online resources ranging from dictionaries to complete machine translation systems. The languages benefiting from such exhaustive treatment however tend to be restricted to the European Family, most notably English and French.

More recently, however, the feasibility of data driven approaches in the context of today’s computing power holds out hope for the rest of us. These are those of us concerned with less studied languages, who have few or no linguistic resources to assist us, and for whom the cost of building these up from scratch is prohibitive.

* Work reported herein was carried out at INRIA, France, supported by the European Research Consortium on Informatics and Mathematics (ERCIM).

1.1 Motivation

Sinhala is a language of around 13 million people living in Sri Lanka. It is not spoken in any other country, except of course by enclaves of migrants. Being a descendant of a spoken form (Pali) of the root Indic language, Sanskrit, it can be argued that it belongs to the large family of Indo-Aryan languages. Tamil, the second most spoken language in Sri Lanka with about 15% of its 18 million people counting as native speakers, is also spoken by some 60 million Indian Tamils in Southern India*. The dialects of Indian Tamil however differ significantly to those of Sri Lankan Tamil to cause difficulty in performing certain linguistics tasks [1]. Though originating in India, it does not share the Indo-Aryan heritage of Sinhala, but rather is one of the two main languages in another (unrelated) family known as Dravidian. Over the past few decades, the people groups using these two language-cultures within Sri Lanka have got so polarized that the prevailing ‘ethnic’ problem is by far the biggest issue facing the country at present. In this setting, any work that is able to provide any semblance of automatic translation between the two languages in question becomes a potential instrument of peace. English is often used in Sri Lanka as a ‘link’ language and also holds the status of an official language together with Sinhala and Tamil.

1.2 E-Readiness of Languages

By far the most studied languages in terms of descriptive as well as computational models of linguistics have been the European ones – among them English receiving more attention than any other. From lexical resources, part-of-speech taggers, parsers and text alignment work done using these languages, fairly effective example-based and statistical machine translation efforts have also been made.

In contrast, electronic forms of Sinhala became only possible with the advent of word processing software less than 2 decades ago. Even so, being ahead of standardizing efforts, some of these tools became so widely used, that almost 2 years after the adoption of the ISO and Unicode standard for Sinhala (code page 0D80), there is still no effort to conform to it. The result is that any effort in collecting and collating electronic resources is severely hampered by a set of conflicting, often ill-defined, non-standard character encodings.

The e-readiness of the Tamil language lies somewhere in between the two extremes of those of English and Sinhala. Here the problem seems to be more the number of different standardization bodies involved, resulting in a plethora of different standards. TASCII was one of the earliest attempts and is an 8-bit Tamil encoding, while the newer Unicode standard is at code page 0B80. Owing to this, available electronic resources may be encoded in unpredictable ways, but tools for translating between non-standard encodings and the standard(s) exist**.

* There are also significant Tamil speaking communities in countries in the far East such as Malaysia and Singapore.

** The author, however, is unable to evaluate the effectiveness of such conversion.

1.3 Scope of Paper

The aim of the present research was to start with a parallel tri-lingual text resource in the Sinhala-Tamil-English language triplet. Several sources in Sri Lanka (academic, press, and the media among others) were pursued without success.

Next the more realistic task of collecting a Sinhala-English corpus was pursued, driven by the fact that the author has working knowledge of both these languages. While many promising lines of inquiry were pursued, owing to reasons ranging from copyright to inability to locate the relevant electronic version, this too had to be abandoned.

A final attempt was made then to search for at least a bi-lingual parallel text in Sinhala-English on the Internet. Many interesting ‘matches’ were found, but upon closer examination, many of these bi-lingual articles and reports were not really ‘translations’ of each other, but rather, those written by two independent writers witnessing or reporting on a single event.

The two successful ‘hits’ resulting in the above search turned out to be those of the World Socialist Web Site and the Mettanet. Both sites have Sinhala-English translations that appear to be fairly consistent with the former adding Tamil and the latter Pali (one of the roots of Sinhala) on an on-going basis.

Owing to these reasons, and in order to test the data-driven paradigm using at least one extensively studied language (English), the scope of the present paper was restricted to the WSWWS Sinhala-English parallel corpus.

2 Machine Translation from Scratch

Since the ultimate goal is to translate between languages without access to prior linguistic knowledge, the primary approach explored was the statistical machine translation models initially proposed by Brown et.al. of IBM in the early 1990’s [2]*.

While ideally a word-aligned parallel corpus would be able to provide much of the data needed to build these models, it is precisely this that is not available in large quantities. The IBM models (referred to in the literature using numbers 1 through 5 denoting the simplest to the most complex) were designed particularly to solve this ‘chicken-and-egg’ problem. Using a bootstrapping agenda, the simpler models (1 and 2) are used successively to build the more sophisticated translation models 3, 4 and 5. More recently, Al-Onaizan et.al. [3], have improved the efficiency of this bootstrapping process and made available public domain tools to build translation models from scratch**.

* While other data-driven approaches such as EBMT exist, the size of corpus they require is often even larger.

** Available for download as the EGYPT toolkit at www.clsp.jhu.edu/ws99/projects/mt/toolkit

2.1 Text Alignment – The Precursor to Statistical Machine Translation

A pre-requisite for building a translation model in this statistical approach is a sentence-aligned parallel corpus – also called a bitext. There are many techniques in the literature for aligning a parallel corpus (e.g. [4]). Many of these techniques rely on sentence length or cognate matching to assist in the process of aligning a bitext. Owing to Sinhala orthography being non-Latin-based, the cognate approach is not obvious, while the sentence length approach is not foolproof.

Melamed [5] describes a much more robust, less language dependent approach, which can use a combination of alignment cues ranging from cognates to part-of-speech tags and translation lexicons. His SIMR/GSA algorithm, however, itself requires a ‘training set’ of aligned sentences from which to learn. Owing to the lack of cognates and POS-tagged corpora for Sinhala-English alignment, this task too was carried out using a bootstrapping agenda.

A very limited translation lexicon based mainly on proper names was hand constructed semi-automatically using simple frequency counts in articles making up the corpus. In addition, two-thirds of the corpus was hand-aligned sentence-wise. Using this incomplete translation lexicon for building an ‘axis generator’ out of the corpus, the remaining one-third of the corpus was aligned.

A very simple algorithm, which treats each word of a sentence in one language as being potentially the translation of each word of the corresponding aligned sentence, was employed for building up a lexicon automatically.

2.2 N-gram Language Models

Apart from building a translation model, the statistical approach also requires access to a language model typically based on n-grams extracted from a monotext. While alternative approaches use POS tags or class-based statistics, in the absence of these, the CMU-Cambridge Toolkit [6] can use minimally tagged plain text corpora for building such n-gram language models*.

This part of the model building process is somewhat independent of the rest of the statistical machine translation approach and so can benefit from much larger monotexts which may provide better language models.

3 The Bootstrapping Agenda

Based on the above plan, we set up the following agenda for building up a ‘bootstrapping bitext’, in order to use it in turn to build a better translation model for Sinhala-English machine translation. The agenda itself is not dependent on the language pair experimented with here, and so, can be used as a general scheme for experimenting with any new language pair with a view to building real statistical machine translation systems.

* Downloadable from svr-www.eng.cam.ac.uk/~prc14/toolkit.html

1. Extract ‘clean text’ – stripping/converting markup of ‘raw electronic form’
2. Extract ‘rough’ lexicon from (1)
3. Use (2) to align bitext
4. Use aligned bitext to bootstrap translation process using IBM models 1 through 4
5. Use each ‘side’ of the bitext for building monolingual language models needed by the decoder

All of the above processes provide better results with improved input. An important part of such ‘improvement’ is the quality of the lexicon. At various stages this agenda provides points of ‘re-entry’ that allow the successive steps to produce better results.

For instance, the ‘rough lexicon’ with which the sentence alignment algorithm is initially fed (2) can be improved with a better lexicon resulting from the sentence alignment process itself (3).

Again, the higher scoring word-aligned sentences output from the IBM models (4), provide word-alignments that can be used to build a better input lexicon for (3).

While this process successively improves the translation model and is the goal of the work reported here, the final objective is to provide a method of translating new (unseen) text in one language to another. In order to achieve this, we need to turn to another device called a ‘decoder’ (using speech recognition terminology). This paper does not cover that process, but German et.al. [7] compare algorithms for decoding, and a public domain decoder called Rewrite at ISI* provides a way to ultimately test out the effectiveness of the models developed using the above agenda.

4 The Sinhala-English Case Study

A set of WSWS articles available on the site during the summer of 2001, which had mutual translations in Sinhala and English, were selected to form a small parallel corpus for this research. This consists of news items and articles related to politics and culture in Sri Lanka.

In attempting to ‘clean up’ unnecessary tags, it was further discovered that owing to the bad design of the character map, some of the characters had actually been encoded using a ‘supplementary font’. In order to simplify further processing, and since there were only a few of such characters, these were simply ignored as they could be readily inferred from context.

After cleaning up the texts, a total of 2063 sentences of Sinhala text corresponding to 1935 sentences of English text were marked up in accordance with TEI-Lite guidelines. This amounted to a Sinhala corpus of 49k words and a parallel English corpus of 52k words. The aligned corpus has 2050 sentences.

* Available for download at www.isi.edu/licensed-sw/rewrite-decoder

4.1 Basic Processing

Word counts for Sinhala and English were extracted from the respective corpora without any morphological processing. This resulted in a Sinhala ‘lexicon’ of 6285 unique words and an English ‘lexicon’ of 6150 words.

In addition, in order to perform multilingual work, part of the bitext was separated out and manually aligned at sentence level. This process brought to light some of the problems of the Sinhala encoding which broke many a text editor. Most of these problems needed to be corrected using semi-automatic search-replace processing later.

4.2 Language Modeling

Owing to the lack of lemmatizers, taggers etc. for Sinhala, all language processing done used raw words and were based on statistical information gleaned from the respective ‘half’ of the bi-lingual corpus. The CMU-Cambridge Statistical Language Modeling Toolkit (version 2) was used to build n-gram language models using both the Sinhala and English ‘halves’ of the corpus independently.

Table 1 shows some statistics of the resulting language models with respect to a small test corpus extracted of new articles on the WSWs site. The perplexity figure for Sinhala is higher than for English. However, in both cases larger test sets produced higher percentages of out of vocabulary (unknown) words indicating that the basic corpus size needs enhancing.

Table 1. Perplexities and other statistics for the Sinhala and English WSWs corpora

Description	Sinhala Corpus	English Corpus
Size of Testset	2667 words	2992 words
Perplexity	509.38 (8.99 bits)	181.46 (7.5 bits)
# 3-grams	349 (13.09%)	800 (26.74%)
# 2-grams	696 (26.10%)	952 (31.82%)
# 1-grams	1622 (60.82%)	1240 (41.44%)
# unseen words	426 (13.77%)	244 (7.54%)

For the purpose of building better language models needed for the statistical translation process a monolingual English corpus was extracted from the same source consisting of 83k words.

4.3 Text Alignment

Much multilingual processing requires that parallel texts be aligned at sentence level. Many of the techniques for automatically aligning parallel texts are based on sentence length and/or identifying cognates in the two languages. In order to make minimum assumptions about the lengths of sentences or words, about their orthography and other ‘hidden’ assumptions in many of these techniques, Melamed’s GSA algorithm

[5] was employed to align the Sinhala and English segments of the corpus. The algorithm is able to use punctuation, cognates and translation lexicon cues in its quest to find potential sentence alignments. Owing to the difficulty in designing an effective transliteration scheme for discovering cognates, a simple translation lexicon was semi-automatically induced from the data.

Several automatic schemes were also explored for extracting translation lexicons, based on the simple assumption that any word in a given sentence of one side of the bitext can correspond to any of the words in a 'window' of the corresponding aligned sentence in the other half of the bitext. Several 'window widths' and frequency thresholds were experimented with and gave results varying from high precision, low recall (80%, 12%) for 'small window', high threshold; to low precision, high recall (35%, 60%) for 'full window', low threshold.

Since the GSA algorithm only requires some cues, and does not rely on a complete translation lexicon, the choice of the scheme for extracting the lexicon was not crucial. Hence the high precision small translation lexicon was used for this purpose. Unfortunately, the optimizing of SIMR parameters for Sinhala was not successful, so that the GSA algorithm had to rely on a non-optimal model.

Based on the hand-aligned two-thirds of the corpus, the algorithm aligned the rest of the corpus with an accuracy of 62% compared to the hand-aligned 'held out' portion of the corpus.

One of the reasons for the lower accuracy turned out to be a large article in the last one-third of the bitext that appears to have been translated by a different source or means. This article was later hand-aligned in order to maintain the very moderate size of the overall bitext.

4.4 Towards Machine Translation

It is clear that 'improving the input' is one of the main tasks ahead. This is no simple task. It has two main dimensions – the quantity and quality.

Some improvement in the former is already underway with much greater access to the archives of WSWs. This is also expected to add the breadth needed to the research by providing Tamil parallel text to the existing bitext. This in turn is expected to give rise to better techniques for fine-tuning the processing as three languages can be expected to give better clues for alignment than two [8].

Improving the quality is much less straightforward. Owing to the 'minimal reliance on linguists' aim of the research, the main approach explored was to feed the system back with an improved translation lexicon.

The long-term goal of the research is machine translation. With this in mind, and given the lack of quality data in any sufficient quantity, statistical machine translation techniques cannot be expected to perform even at the relatively modest state-of-the-art levels with which it performs on massive English-French bitexts. Nevertheless, for the sake of deciding on the suitability of these techniques on virtually un-related language pairs, the Sinhala-English bitext was subjected to statistical machine translation techniques proposed by the 'IBM models'.

The models proposed are compositional, in that they use monolingual statistical language models of each of the constituent languages of the bitext, together with very simple word-based translation models based on the sentence aligned bitext to build decoders that are able to translate new (unseen) sentences.

The EGYPT toolkit of JHU [3] was used to build IBM model 3 type translation models for Sinhala-English translation. A byproduct of this process is the output of a word-aligned form of the input bitext, which is one means of assessing the success of the translation model learned from the bitext.

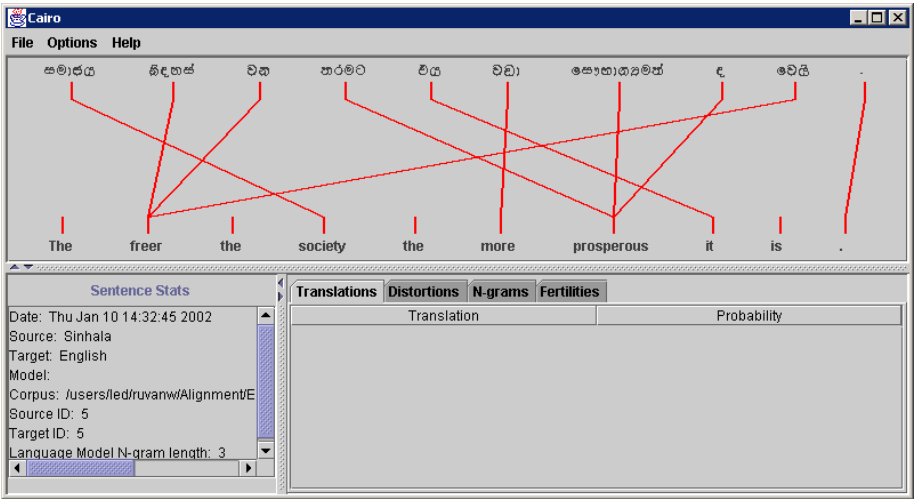


Fig. 1. A word-based alignment visualization tool ‘customized’ to display Sinhala characters

Figure 1 displays (using Cairo, the alignment visualization tool in EGYPT) a typical word-based alignment inferred from the bitext. We can see from it, for instance, that while the algorithm has correctly learned that ‘සමාජය’ corresponds to ‘society’ and ‘වඩා’ to ‘more’, it has mistakenly learned that ‘තරමට’ and ‘සෞභාග්‍යමත්’ correspond together to the word ‘prosperous’ (even though ‘සෞභාග්‍යමත්’ alone means ‘prosperous’).

4.5 Bootstrapping

In order to improve the input to the process, two sources were explored. The first is to extract word-mapping information gleaned by the text alignment process and the second to that learned by the translation model.

The GSA algorithm learns to mainly distinguish between multiple translations of a particular word in one language within a given context. As such there is no direct method for enhancing a translation lexicon from the process of alignment. However, work is now underway to use the basic lexicon extraction strategies outlined in 4.3, with better ‘window’ information extracted from GSA’s output, in order to feed the translation lexicon building process.

The translation model on the other hand produces direct word-alignments with associated probability scores. Initial attempts to use these scores to filter out incorrect alignments show that though the quality of the translation lexicon (in terms of precision) cannot be improved, its size can be, by incorporating the proposed word-alignments.

The following example demonstrates how such ‘feedback’ processing improved the final word-alignment produced by the translation model.

Alignment score : 3.86678e-11

සෘම මිනිසෙකුටම ඔහුගේ ම ශ්‍රමය තුළ දේපලක් විය .

NULL ({ 9 }) Every ({ 1 }) man ({ 2 7 8 }) had ({ }) a ({ }) property ({ }) in ({ }) his ({ 3 }) own ({ 4 }) labour ({ 5 6 }) . ({ })

Alignment score : 1.56548e-06

සෘම මිනිසෙකුටම ඔහුගේ ම ශ්‍රමය තුළ දේපලක් විය .

NULL ({ }) Every ({ 1 }) man ({ 2 7 8 }) had ({ }) a ({ }) property ({ }) in ({ 6 }) his ({ 3 }) own ({ 4 }) labour ({ 5 }) . ({ 9 })

The improved model (below) has correctly mapped ‘ශ්‍රමය’ to ‘labour’ and ‘තුළ’ to ‘in’ instead of mapping them both to labour as in the initial alignment.

5 Remaining Problems and Conclusion

Among the problems identified in the shortcomings of the present statistical language and translation models to handle the virtually unrelated languages, are (a) the limited size of the corpora highlighted by the high perplexity figures in the training corpus, (b) the insufficient ‘signal’ produced by lexical cues for alignment and possibly (c) the long-distance ‘movement’ of mutually translated words and phrases not captured in the translation models.

In order to address (a), current efforts are underway to extract a larger corpus from the WSW site. In addition, some initial work with the larger Mettanet corpus is to be undertaken. For (b), a set of possible enhancements are planned to ‘amplify the signal’. These include the use of numeric data in the corpus as potential points of correspondence, the development of a phone-based cognate identification system, the use of lemmatizing of words and exploring class-based models in order to have more data through aggregation. If the improved language and alignment models fail to achieve better results in terms of statistical machine translation between such ‘unrelated’ language pairs as suggested by (c), serious consideration would need to be given to the underlying assumptions of the IBM models being pursued.

The main result of the bootstrapping agenda proposed here is the availability of a Sinhala-English lexicon and a small, aligned Sinhala-English bitext. These in turn will act as ‘seeds’ for building more wide coverage translation lexicons and larger aligned bitexts. Apart from this, adding a third language, Tamil, into the project is expected to be carried out in an analogous manner, in order to construct the desired ‘tritext’. Indeed, preliminary collecting and collating of the parallel Tamil text indicate that the primary goal of this work, namely the translation between Sinhala and

Tamil, may prove to be more forthcoming with better results than is possible with the pair Sinhala-English. The final goal is to be able to complete the statistical machine translation process by building decoders from the language and translation models built this way for the relevant language pairs.

It is clear that any real-world automatic language translation effort will benefit from linguistic knowledge. The aim of this research is to determine how far non-knowledge intensive methods can be extended towards achieving success in the context of languages for which such resources are scarce or non-existent.

Results obtained herein using techniques employed largely in the processing of European languages, show some promise that, with more work, effective language and translation models for 'new' language pairs could be built using publicly available tools in the not-too distant future.

Acknowledgements: The author wishes to express his gratitude to the European Research Consortium on Informatics and Mathematics (ERCIM) for providing a grant to pursue the above work at INRIA, Lorraine in France during 2001/2002. He is also grateful to the University of Colombo for releasing him during this period from his regular duties.

References

1. Germann, U.: Building a Statistical Machine Translation System from Scratch: How Much Bang Can We Expect for the Buck. Proceedings of the Data-Driven MT Workshop of ACL-01. Toulouse, France (2001)
2. Brown, P. F., Della-Pietra, S. A., Della-Pietra, V. J. and Mercer, R. L.: The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2) (1993) 263-311.
3. Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F.-J., Purdy, D., Smith, N. A., and Yarowsky, D.: Statistical Machine Translation, Final Report, JHU Workshop 1999. Technical Report, CLSP/JHU (1999)
4. Gale W. A. and Church K. W.: A program for aligning sentences in bilingual corpora. Proceedings of ACL-91, Berkeley (1991) 177-184
5. Melamed I. Dan: A Portable Algorithm for Mapping Bitext Correspondence. Proceedings of the 35th Conference of the Association for Computational Linguistics (ACL'97), Madrid, Spain (1997)
6. Clarkson, P.R. and Rosenfield, R.: Statistical Language Modeling using the CMU-Cambridge Toolkit, Proceedings ESCA Eurospeech, Rhodes, Greece (1997)
7. Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K.: Fast Decoding and Optimal Decoding for Machine Translation. Proceedings of ACL-01. Toulouse, France (2001)
8. Simard, M.: Text-translation Alignment: Three Languages Are Better Than Two. In Proceedings of EMNLP/VLC-99, College Park, MD (1999)

A Report on the Experiences of Implementing an MT System for Use in a Commercial Environment

Anthony Clarke¹, Elisabeth Maier², and Hans-Udo Stadler¹

¹ CLS Corporate Language Services AG, Elisabethenanlage 11, CH-4051 Basel, Switzerland
anthony.clarke@cls.ch, hans-udo.stadler@cls.ch
www.cls.ch

² Canoo Engineering AG, Kirschgartenstr.7, CH-4051 Basel, Switzerland
elisabeth.maier@canoo.com
www.canoo.net, www.canoo.com

Abstract. This paper describes the process of implementing a machine translation system (MT system) and the problems and pitfalls encountered within this process at CLS Corporate Language Services AG, a language solutions provider for the Swiss financial services industry, in particular UBS AG and Zurich Financial Services. The implementation was based on the perceived requirements of large organizations, which is why the focus was more on practical rather than academic aspects. The paper can be roughly divided into three parts: (1) definition of the implementation process, co-ordination and execution, (2) implementation plan and customer/user management, (3) monitoring of the MT system and related maintenance after going live.

1. Introduction

This paper describes the implementation of an MT system in a commercial environment and the steps which are necessary to set up efficient and cost-effective processes in order to do so. The work described in this paper follows up on the MT system evaluation reported in [1].

The system, DTS/Globaliser from Compendium (formerly SailLabs), is currently being operated at CLS Corporate Language Services AG for all language directions between German, English and French. The system has been tailored to the translation of texts from the finance and insurance domain. Extensive language resources developed at CLS Corporate Language Services AG, such as translation memories and multilingual term banks, have been used to bootstrap the system in a short amount of time.

It is hoped that this paper will contribute to the discussions about the commercial viability of MT systems and provide both manufacturers and users with some useful insights into the pros and cons of implementing such systems.

2. Definition of the Implementation Process, Co-ordination and Execution

The process of the implementation had to be determined and co-ordinated by a central team, which of course had to be built up; this process comprised

- the development and set-up of an infrastructure in which the machine translation project can be hosted. This subsumes both the building up of an MT and the development of a scalable IT infrastructure;
- the acquisition and management of MT users and the development of an MT cost model, including the development of a plan for the introduction of Machine Translation at the customer site;
- the bootstrapping of the first MT release and, subsequently, implementation of scalable releases, which respond to the requirements of the MT users.

2.1 Infrastructure Development

Team

It first had to be decided how big the team of employees was going to have to be, in order to assure a smooth bootstrapping and operation of the MT system.

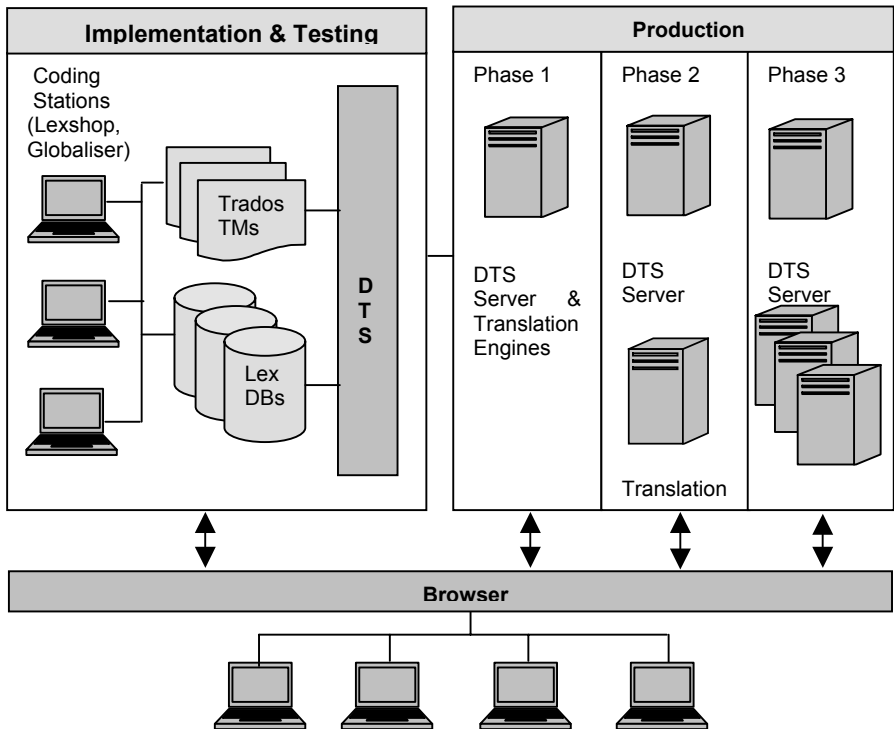


Figure 1: Scalable IT infrastructure for MT at CLS Corporate Language Services AG.

At CLS this was defined as 3 full-time employees and 6 part-time staff accounting for another 3 FTEs plus 2 outside consultants. After an initial phase which was mostly dedicated to tasks like team building, team education, infrastructure set-up and bootstrapping of linguistic resources, the emphasis of the work shifted towards more operational tasks, e.g. user-driven extension of the linguistic resources and adaptation of the system, process refinement and infrastructure extension.

IT Infrastructure

For a reliable operation of the MT system a dedicated server infrastructure was set up (see Figure 1). This infrastructure includes

- A server for testing new configurations and data before the launch of a release;
- A scalable range of production servers which can be flexibly adapted to customer requirements and to the overall translation volume;
- Monitoring mechanisms to increase the overall stability of the service and to guarantee the minimization of system downtimes;
- A testing framework to determine the system behaviour under certain load scenarios in order to adapt the server infrastructure to changing demands;
- A set of coding stations, including a sophisticated MT lexicon editor (Lexshop) for lexicon entry and maintenance, and a desktop version of the translation system (Globaliser) for testing the effects of lexicon changes.

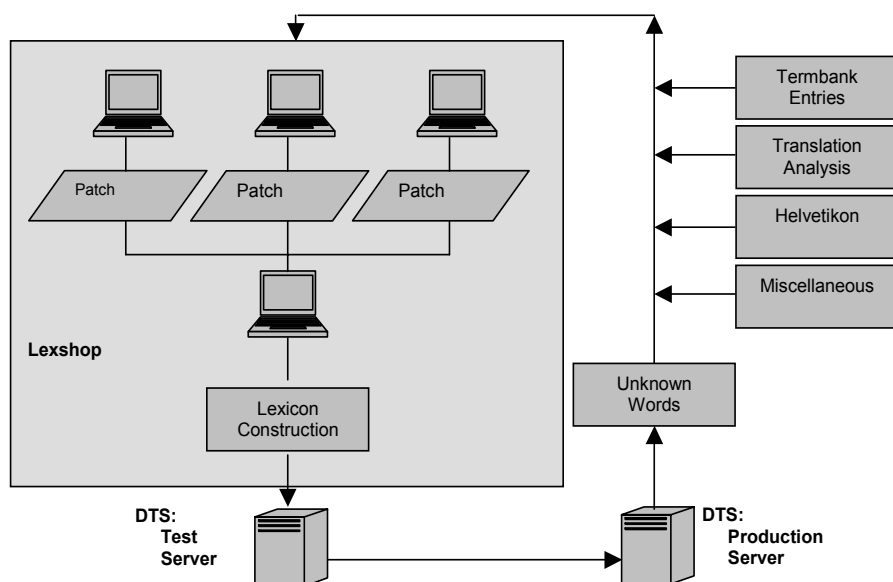


Figure 2: Work process in the MT Team.

The process of implementation and testing, i.e. of gradually extending the Machine Translation system in order to respond to customer needs, to render translation processes more time and cost efficient and to continuously improve translation quality, is shown in Figure 2.

3. Implementation Plan and Customer Management

As described in [1] it was planned to introduce the MT system in three phases with a gradually growing customer base:

- 20 MT users, focus on one language pair, one customer;
- 200-300 MT users, focus on all available language pairs, one customer;
- Full rollout targeting many thousands of users from various customer segments.

This method was chosen in order to allow a gradual growth of the MT system and the IT infrastructure while, at the same time, allowing the MT staff to gain experience with the new technology. At the time of the writing of this report, the second project phase had been successfully concluded.

3.1 The First Project Phase

Due to problems with the reliability of the service and with the incompleteness of the vocabulary, mixed feedback was provided: on the one hand, negative comments were made about the usefulness and availability of the service, while on the other hand the service continues to be used regularly up to this day: on average, every user who participated in the first project phase uses the MT service once per day. It must be observed, though, that to a large extent the service is used for the translation of very short texts, often only words or phrases are looked up. Table 1 gives an impression of the MT system usage during the first project phase.

Table 1: Usage Statistics of the first project phase

Translation Direction	Document Translations	Text Translations	Total Number of Requests
D-F	50	1586	1636
E-F	9	161	170
F-E	2	116	118
F-D	3	337	340
Total	64	2200	2264

3.2 The Second Project Phase

Set-up

The experiences made in the first project phase were used as input for the set-up of the second project phase. The following measures were taken:

- One person was allocated specifically for support of the user group of the first project phase; this person collected personal feedback about the performance of the system, about wishes for an extension of the system and provided information about future releases, etc.
- Expectation management of the user was identified as a major objective. Much effort was put into the task of informing the future users about the contexts in which MT shows best results. Well before the launch of the second project phase, information was broadcast on the customer's intranet, mails were sent out,

and a telephone support line was set up, where information about the project could be obtained. Most importantly, a questionnaire was sent to the registered users in order to find out about the intended usage situations (language directions, text types, translation quality). Some of the results of the survey are summarized in Tables 2 - 4.

Table 2: User survey carried out before second project phase: translation directions

	D-E	D-F	E-D	E-F	F-D	F-E
Most important translation direction	53.3%	42.4%	22.5%	4.4%	15.2%	1.2%
2 nd most important translation direction	16.4%	21.2%	22.5%	8.8%	18.1%	9.5%
3 rd most important translation direction	7.4%	12.7%	8.5%	13.2%	16.2%	10.7%
4 th most important translation direction	9.8%	10.2%	14.0%	26.4%	16.2%	21.4%
5 th most important translation direction	13.1%	13.6%	32.6%	47.3%	34.3%	57.1%

Table 3: User survey carried out before second project phase: document formats

	.html	.doc	.rtf	.ppt	.txt
Most important translation direction	10.0%	85.5%	1.5%	10.7%	6.1%
2 nd most important translation direction	18.8%	6.1%	8.4%	35.9%	12.2%
3 rd most important translation direction	15.3%	1.5%	6.1%	14.5%	8.4%
4 th most important translation direction	15.3%	2.3%	13.0%	9.9%	13.7%
Less important document formats	42.7%	4.6%	71.0%	29.0%	59.5%

Table 4: User survey carried out before second project phase: expected MT quality

	Perfect	Close to human	Gist preserved
Mentionings	13	68	24

- On the system side, care was taken to increase the usability of the system
 - Help pages were created to describe how the system is used and how the text input can be tuned in order to improve translation quality.

- FAQ pages are maintained on a regular basis, taking user feedback into account, providing information about problems and possible solutions.
- Mail and telephone support is available to the user via various channels.
- A customer database with contact details for every user is available in order to optimise support,
- In all communication with the users, feedback concerning the improvement of the system was strongly encouraged.

System Implementation and Rollout

After the development of an IT infrastructure, the MT system had to be geared towards the translation of texts as produced by the customers of CLS, i.e. in the financial services industry. The vocabulary and the writing conventions commonly used in this area had to be coded into the system. This process could be streamlined by a re-use of the linguistic resources available at CLS: translation memories and multilingual term banks. In the following sections we describe how these data can be incorporated into and used by the machine translation system.

Use of Translation Memories

For a number of years, all translations produced by CLS Corporate Language Services AG have been incorporated into TRADOS translation memories. DTS/Globaliser, on the other hand, can exploit TRADOS translation memories insofar as translation segments already available in the memory can be inserted directly into the translation result, without going through the machine translation process. Since all segments in the translation memory undergo strict quality control, the re-use of this information increases the overall quality of the translation result.

TRADOS translation memories are integrated into DTS/Globaliser by exporting the memory data into a predefined format and by subsequently importing them into the MT system. This process, although it has to be done manually, can be done in a number of hours. In one of the future system releases it will be possible to have the memory linked to the machine translation system, so that regular synchronizations with the company's translation memory can be avoided.

Terminology Import

In order to ensure the consistency of the terminology across all services of CLS Corporate Language Services AG, it had to be ensured that the customer vocabulary, which is available in Multiterm term banks at CLS, was also incorporated into the machine translation system.

Since much of the information needed by a machine translation system is not available in a term bank, as these were developed for (human) vocabulary lookup instead of automatic processing, missing information had to be added before the vocabulary could be imported automatically. This information concerns, for example, semantic features, word categories (where missing), various types of grammatical information, etc. Given the size of the CLS term bank, this task was very time consuming and was therefore outsourced to SailLabs, who had a toolbox to facilitate this task.

The import of the term bank entries could be automated in cases where

- entries were complete, i.e. where both source and target entries were available in the term bank together with their corresponding category information.
- category information missing in the term bank could be retrieved from the SailLabs lexicon.
- entries were labelled as *abbreviation* in the term bank.

In all other cases manual editing was required.

Results of the Second Project Phase

A survey carried out after the second project phase showed very positive results: well over 70% of the users participating in the survey said that the machine translation tool facilitated their everyday work. Close to half of the users think that the discontinuation of the service would have noticeable side-effects. While most of the users considered the vocabulary as satisfactory, an improvement of style and grammatical correctness was high on their wish list.

The survey showed that the service was being used for the pre-translation of documents, followed by word-to-word translations (dictionary lookup) and for text comprehension. Looking at the most frequently used translation directions (German-English and German-French, which were both requested in about 28% of the cases) we conclude that the service is frequently used by employees, who need to understand internal German communication but who do not understand the language sufficiently. This fact is further supported by the finding that, rather than finance or IT texts, general texts are the most frequently translated. The results of the survey will be described in detail in a forthcoming paper.

It has to be mentioned that knowledge about the service, although initially only provided to subscribed users, has been spread by mouth-to-mouth propaganda.

4. Testing and System Availability

Before the launch of every new release, load tests were carried out to simulate the expected translation load on the intended server architecture. In order to do so, a load-testing environment, provided by the SailLabs development team, was used.

An important user requirement was the 7x24-availability of the translation service. In order to guarantee high reliability, monitoring services have been implemented which alert the support staff about unrecoverable situations. Recovery scripts have been put in place as well.

We also had to take various security aspects into account, such as the separate administration of different client data, even to the extent of using separate servers, and the provision of secure channels for the transfer of data.

Due to potential proxy problems, the access to the system had to be tested from various customer locations well before the launch of each release.

Generally speaking, the monitoring and related maintenance of the system should be given high priority because, as has been shown in our case, the issue of document conversion can be a problem, jeopardising the availability of the system. And a system which is unavailable for a lot of the time will soon lose the interest of the users.

Outlook

As a next step, the translation service will be rolled out in UBS AG. It will be available at every desktop as part of the UBS intranet. Currently, an accounting infrastructure is being setup, which will allow tracking of the translation volume.

Acknowledgements

We would like to thank our colleagues at CLS Corporate Language Services AG, at Canoo Engineering AG, and at CSF AG for their help with the Machine Translation system. We would also like to thank the colleagues at Compendium for their friendly and competent support.

References

1. Maier, E., Clarke, A., Stadler, H.-U.: Evaluation of machine translation systems at CLS Corporate Language Services AG, Proceedings of MT Summit VIII, Santiago de Compostela, Spain (2001) 223-228.

Getting the Message In: A Global Company's Experience with the New Generation of Low-Cost, High Performance Machine Translation Systems

Verne Morland

NCR Corporation, Global Learning Division, WHQ-3, 1700 S. Patterson Blvd.,
Dayton, Ohio, U.S.A. 45479

Abstract. Most large companies are very good at “getting the message out” – publishing reams of announcements and documentation to their employees and customers. More challenging by far is “getting the message in” – ensuring that these messages are read, understood, and acted upon by the recipients. This paper describes NCR Corporation's experience with the selection and implementation of a machine translation (MT) system in the Global Learning division of Human Resources. The author summarizes NCR's vision for the use of MT, the competitive “fly-off” evaluation process he conducted in the spring of 2000, the current MT production environment, and the reactions of the MT users. Although the vision is not yet fulfilled, progress is being made. The author describes NCR's plans to extend its current MT architecture to provide real-time translation of web pages and other intranet resources.

1 Introduction

NCR Corporation, headquartered in Dayton, Ohio, is a global technology solutions and services company with 32,000 associates in 80 countries. NCR has five major business units and hundreds of professional and technical job roles. Over 50% of NCR's workforce reside outside the United States.

Like most modern companies, NCR is very good at “getting the message out” – publishing reams of announcements, brochures, instructions, and other documents to their employees and customers. More challenging by far is “getting the message in” – ensuring that these messages are read, understood, and acted upon by the recipients.

Although English is the official language of the company, many associates are not fluent in English. This impairs not only their ability to read company documents and converse with their English-speaking colleagues; it also makes it more difficult for them to stay abreast of global company developments and even to take advantage of specific opportunities, for example, training programs, that would help them improve their performance. In short, it reduces their productivity.

NCR also generates more than half of its revenue outside the United States. When country-level revenue statistics are combined on the basis of local language, it is clear

that effective communication in Japanese, German, French, Spanish, and Italian is very important to the company.

In recent surveys of NCR associates around the world, between 10 and 15 percent of respondents say they would prefer to receive company communications in a language other than English. The actual numbers are probably much higher due to several factors. The most important of these are: 1) since the surveys were in English respondents tend to be those who are comfortable with English, 2) many respondents may not have taken the question seriously since they may have assumed that a vote for anything other than English would have no practical impact, and 3) respondents were disproportionately drawn from the upper echelons of the company who have better training in and more exposure to English.

One verbatim response to the question, "In which language would you prefer to receive your company communications?" was telling.

"Although English is an international language, it should be that the information is agreement to the country. So that it is but understandable the information and we all are speaking of the same communication. It is ridiculous that they worry about the communication without everybody speaks English."

In both style and content this is a great argument for the use of good translation services...

1.1 NCR's MT Vision

In January 1998, an advanced technology team in NCR's Organization Development group became aware of the newly released web page translation service offered on the Altavista internet search site using machine translation (MT) technology from SYSTRAN Software, Inc. This team monitored the new developments in MT technology as it migrated from mainframes to minicomputers to PCs and as the software moved from highly customized, bespoke systems to shrink-wrapped, off-the-shelf packages.

Excited by these advances, the author formulated this simple vision.

NCR should have a global company intranet on which associates can navigate in the language of their choice and have the entire contents of that intranet appear to be in that language.

1.2 A Scenario Using Transparent MT on a Global Intranet

To understand the impact of this vision on a practical level, consider this scenario.

- 1 An NCR associate in France launches her web browser and enters the URL of the page on the company intranet (or public internet) she wants to see.

- 2 The first time she does this she is connected automatically to a central NCR translating web server that presents a screen that says (in French), "What language would you like to use?" She chooses French. [This preference would be stored in a web "cookie" on her PC so that it would not be necessary for her to make this selection again.]
- 3 The translating server (acting as a proxy server) then retrieves the requested page, translates it, and sends it to the French user's browser which displays it.

Up to this point the user-system interaction and result are similar to the way some free internet translations services now operate, but the scenario continues.

- 4 The French user views the page she requested in French. The page contains a number of links to other information. She clicks on one of these links.
- 5 The user's request is sent back through the translating server; it does not go directly to the server originally specified on the original source (untranslated) web page.
- 6 When the translating server receives the request for a new page, it knows that the request came from a user who wants to see French. So without stopping to ask the user again what language she wants, the translating server retrieves the requested page, translates it, and sends it to the user.

Now this is different from the services commonly available on the internet. In this scenario, the user has the feeling that she is simply surfing the company intranet and every page is coming up in French. She is using no special software on her PC, just a standard internet browser.

[Note: The scenario described above was created four years ago and is a simplification of the actual implementation that would be used today. It suffers several technical deficiencies, but these do not detract from the purpose of this illustration.]

This technique of transparent translation will ultimately improve productivity by eliminating the mental clutter that would otherwise be introduced by constant translation requests. The key to its success is not only the technical challenge of embedding the translation server invisibly between the user and the source information. Equally important is the quality of the translation.

In the example above, the illusion of a French intranet can only be preserved if the translations are good enough to permit the rapid "scan-click, scan-click" behavior that has become widely known as "web surfing." If the user stumbles over every other sentence or, worse, cannot fathom the meaning of a particularly inept turn of phrase, the spell is broken and the productivity gains are lost. This, then, is the challenge today: to implement the right level of MT technology in the right context, in other words, for delivering the right source texts to the right audiences.

2 Selecting an MT System

In January 2000, NCR invited two MT suppliers – Lernout & Hauspie of Brussels, Belgium, and SYSTRAN Software of Paris and San Diego, California - to participate in an MT “Fly-Off” competition. Modeled loosely on the type of competition sponsored by national airforces to evaluate new military aircraft, NCR’s program called for both suppliers to install prototype systems on NCR servers. These systems were similar in cost and capability.

2.1 Selection Criteria

Once installed on NCR servers, these systems were tested by a team of US and international associates and judged on the basis of: 1) quality of translation, 2) simplicity of user interface and system integration, and 3) total cost of ownership.

In judging the quality of translations provided by the MT systems, NCR’s evaluators focused on the clarity and accuracy of meaning and the avoidance of outright errors. Recognizing that MT systems do not yet deliver perfect translations, NCR’s evaluators looked for the system that produced the clearest, easiest to read language that correctly conveyed the gist of the original text. When mistakes were made they assessed whether the mistakes were merely clumsy constructions or outright errors. Instances of the latter detracted more significantly from the scores awarded in this category.

The second criterion - simplicity of the user interface and system integration – reflected the fact that for this technology to be usefully applied, the MT system must be well integrated with the target applications. In the case of the pilot, this was the web server that hosted the NCR University Online Campus (NCRU). NCR wanted the translations to take place behind the scenes and to introduce no significant delay in web information delivery. From the system support standpoint, NCR looked for the MT system to be robust (high availability/high reliability) and to require no extraordinary operator effort or regular intervention.

When the first two criteria were met, the author made a final judgement factoring in the Total Cost of Ownership over three years. Costs included initial license fees, annual maintenance and update charges, and special one-time or on-going support charges, if any.

NCR advised the suppliers that all three criteria had to be met within practical limits in order for NCR to proceed to the production phase of the project, i.e. putting an MT system into continuous use on the NCRU web site.

2.2 The Test Environment

The evaluation took place in May 2000. The test environment consisted of 50 web pages representing typical content from all areas of NCR University. Thirteen NCR

associates in eight countries on four continents evaluated translations into four languages: French, German, Spanish, and Japanese.

The systems were assessed on two levels:

1. **Page Level:** Each of 35 pages from which the suppliers built special NCR dictionaries were evaluated separately. In addition, evaluators were asked to review 15 new pages that contained similar content, but which were not used for dictionary input.
2. **System Level:** When they completed the page level evaluations for both systems, evaluators were asked to complete a short web survey summarizing their impressions and providing a "bottom line" vote on whether either one of the test systems would be practical to implement at that time.

Overall quality on each page was rated on a 1-5 scale (with 5 being best) and serious errors per page were simply counted. The system level evaluation consisted of a 5-question survey. The fourth question (below) addressed the central issue.

Based on this test, what position would you recommend that NCR take on machine translation of web sites for your language?

- a. Use it now throughout the NCR intranet - it works.
- b. Test it with a larger audience in selected areas of the NCRU web site only.
- c. Do not apply it now, but continue to monitor progress in this technology closely.
- d. Don't waste more time - this is still years away from a practical use.

2.3 Fly-Off Results

In the detailed, page-by-page analyses, the evaluators gave a slight edge, quantitatively and qualitatively, to SYSTRAN. On the overall "Go/No Go" question (above), the results were generally unfavorable. None of the evaluators felt that either system was ready for a large-scale deployment and only a couple thought it would be worthwhile to test them with a larger audience on one web site. Most suggested monitoring the technology further and a few believed that no use would be practical for several years.

In a discussion of this negative result one of the Spanish evaluators, an associate in Argentina, made an insightful observation (reproduced here verbatim).

"...I am fluent in English, and can read it effortlessly. (probably this is true with most of the evaluators). So, I surely prefer to read English than bad Spanish. But maybe it is not true for all the people that only reads English with great effort.

Maybe you could find a group of evaluators that need the translations and ask them not is the translation is good (it is not), but wether they would prefer to read the translated version, however bad, rather than the original.”

The project team decided to press on and the results achieved with larger, more randomized audiences support this observation (more on this in Section 5.)

In October 2000, NCR purchased the Enterprise Translation System from SYSTRAN. The order included the “engines” for translations between English and French, German, Italian, and Spanish. As a result of the collaboration between NCR and SYSTRAN during the fly-off project, SYSTRAN’s server software included a newly developed plug-in for Microsoft’s IIS™ web server. This plug-in was designed to monitor web page requests and route selected pages through the translation engines. The basic elements of the configuration are illustrated in Figure 3 in the next section.

3 NCR’s Initial Plan for MT on the Web (Real-Time Translation)

NCR’s original plan called for pages to be automatically translated for all visitors who expressed their preference for a language other than English. If they got something they couldn’t understand or that looked odd, they could request the original English page.

Since current MT technology produces only rough or “gist” translations from uncontrolled input, NCR decided that it would be very important to advise users that the translations they were reading came from a machine. This notice, embedded in a colorful banner at the top of every page (see Figures 1 and 2), also provided a link to the source text to assist users in the event of poor or mistranslations.

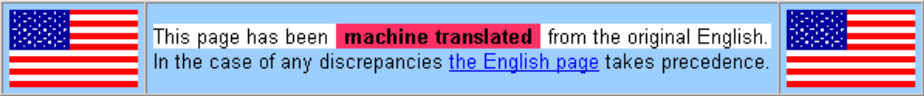


Fig. 1. English Text of the MT Advisory Banner.

For pages translated from U.S. English to French, this banner would appear as illustrated in Figure 2 below.

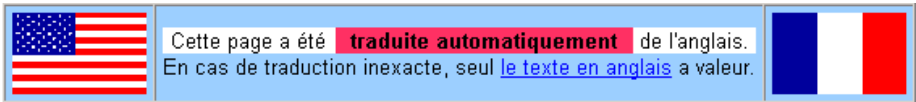


Fig. 2. French Text of the MT Advisory Banner.

The architecture to provide this real-time web page translation service is illustrated in Figure 3 below.

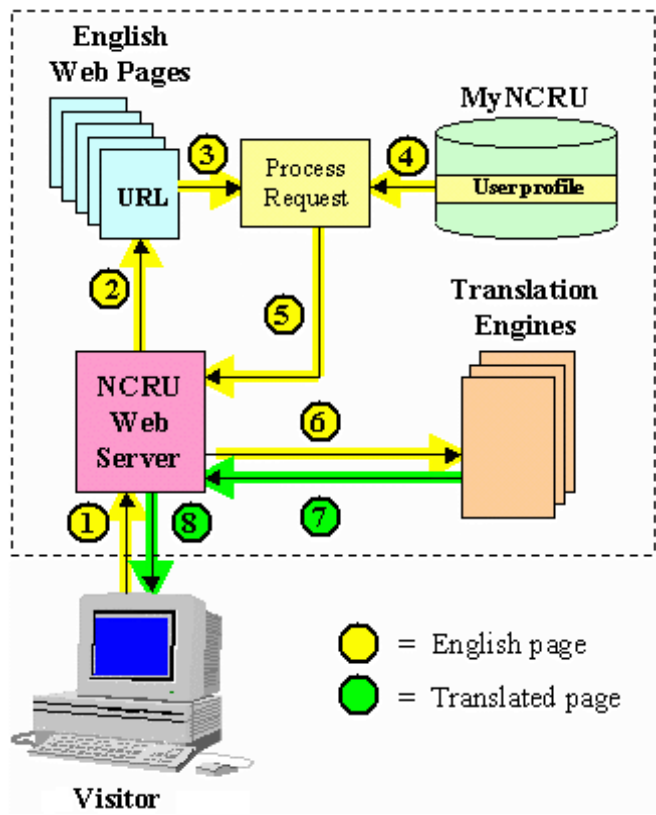


Fig. 3. Architecture for Real-Time Web Page Translation.

Based on the fly-off evaluation, however, none of the NCR evaluators recommended proceeding with a real-time translation system configured in this way. The consensus was that the translations from both suppliers were at best clumsy and at worst wrong and the evaluators felt it would not benefit their local colleagues to have these translations as their default view of the pages on the NCR University web site. Most extreme in this view were the Japanese evaluators who felt that the translations into their language were a long way from being ready for productive use.

One evaluator suggested that NCR take a less ambitious approach. In this scenario users would actively request translations in a method similar to, but more streamlined than, the methods that are offered by many suppliers on the public internet today. All NCR University web pages would be presented in their original English form. If the visitor had trouble understanding the text, he or she could click a button in the header or footer of the page that reads "Translate This Page." Since nearly 2,000 of our 19,000 registered users have indicated which non-English language they prefer, NCR could translate the pages into their languages without having to ask them to supply the language pair each time.

In this second approach the translation would be on demand, rather than automatic. If users felt the translations were not helpful, they would not have to use them. If, on the other hand, NCR discovered through a log analysis that many users were regularly requesting translations, automatic translation could be offered. This would give NCR the quantitative business case needed for a more comprehensive implementation.

Due to technical difficulties connecting the translation engines to the web server, NCR has not put either real-time web page approach into production at this time. The next section describes a batch translation process for pure MT translation that is now in production for HTML formatted newsletters.

4 Publishing a Global Newsletter (Batch Translation)

As a company, NCR's motto is "turning transactions into relationships." This means that NCR assists its customers in converting terabytes of raw data about transactions with their customers (point-of-sale data in retail stores, financial transactions in banks, etc.) into valuable business intelligence that enables NCR's customers to serve their millions of customers more personally and effectively.

Applying this same philosophy internally, NCR's Global Learning division introduced two important personalization services for NCR University – "MyNCRU" web pages and the "MyNCRU Personal Learning News" – a monthly email publication.

The content of the Personal Learning News (PLN) is drawn from an online news and calendar database. The contributors to this database are HR and learning staff members from all over the world. To build a copy of the PLN, the NCRU server compares the keys associated with the most recent news and calendar entries with the set of keys stored in the user's personal profile. It then generates the email using the items that match, sends the message, and moves on to the next subscriber. The current system builds and sends two such personalized messages per second.

Since each of nearly 6,000 copies of the PLN newsletter is individually constructed for its recipient it is not feasible to consider a translation process that would involve human intervention. Using MT software the NCRU server can translate the PLN newsletters from English into French, German, Italian, and Spanish for subscribers who have requested those languages. PLN newsletters are typically 3-4 pages long. These are translated by NCR's current system at the rate of two newsletters per minute. Figure 4 illustrates how translated copies are created, translated, and sent.

The system stores the original English versions of all translated newsletters on the NCRU web site. The preamble to the translated newsletters explains that they are translated entirely by machine and provides a link back to the English originals in the event that subscribers have any questions about or problems with the translation. (The translations of a sample newsletter are available for review on the public internet at this address: <http://www.geocities.com/morlav/pln/translations.htm>.)

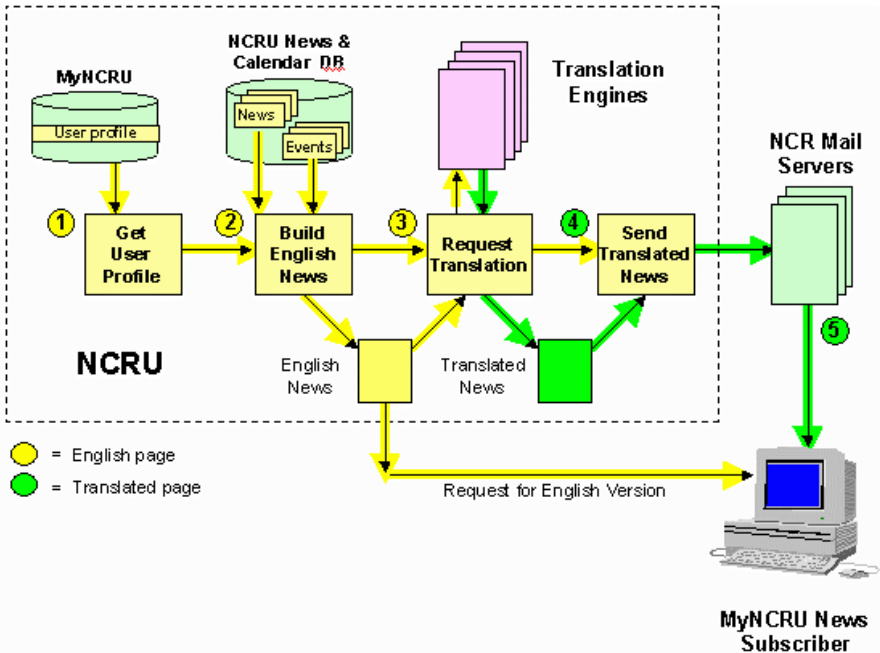


Fig. 4. Process by which the MyNCRU Personal Learning News is created, translated, and sent to subscribers. The yellow line at the bottom of the diagram illustrates how the subscriber can request the original English version via a hyperlink in the translated message

5 **Subscriber Reaction**

In the nine months since the first issue the PLN subscriber base has grown 171% to 5,787. (This is a CMGR – compound monthly growth rate – of 13% per month.) Of these 481 (9%) have requested to receive their copies in French, German, Italian, or Spanish. Since the first issue the requests for translations have grown 152% from 191 to 481. (This is an 12% CMGR.) Since the newsletter’s inception NCR has published 31,066 individual copies of which 2,383 (8%) were machine translated. Only 52 subscribers (less than 1%) have cancelled their subscriptions.

Following the first issue of the PLN, 653 (31%) of the 2,133 charter recipients responded to an 8-question online survey. The overall reaction was very favorable, but response to the machine translation of the PLN was been mixed. When asked to list the three things they liked most about the newsletter, several respondents said: “the translation,” “was translated in French,” and “There is a German version of the Newsletter.” One respondent apparently felt it was now OK to express himself in the language he prefers and answered the entire survey in Spanish.

On the other side, when asked to list the three things they liked least about the newsletter, other respondents said: “automatic translation,” “translation is sometimes funny,” “Translation almost incomprehensible,” and “the translation is very poor.”

Based on these divergent views, NCR’s current hypothesis on the usefulness of pure machine translation is the following.

Those who speak English well will prefer to read English rather than a clumsy and occasionally inaccurate version of their native language. Those who do not speak English well will prefer to read the machine translation and refer back to the original only when they encounter something that appears to be mistranslated. In the second case, the availability of the “gist” translation can significantly improve reading speed and comprehension, thereby increasing associate productivity.

In quantitative terms, the popularity of the translated editions has kept pace with that of the English newsletter.

Table 1. Publication Statistics for the English and Translated Copies of the Personal Learning News

ISSUE	DATE	ENGLISH	TRANS- LATED	%
=====	=====	=====	=====	=====
Vol. 1, No. 1	Jul 2001	2,044	191	8.5%
No. 2	Aug 2001	1,971	195	9.0%
No. 3	Sep 2001	2,172	204	8.6%
No. 4	Oct 2001	2,234	204	8.4%
No. 5	Nov 2001	2,786	277	9.0%
No. 6	Dec 2001	2,948	285	8.8%
Vol. 2, No. 1	Jan 2002	3,075	298	8.8%
No. 2	Feb 2002	3,287	309	8.6%
No. 3	Mar 2002	4,593	420	8.4%
No. 4	Apr 2002	4,727	431	8.4%
		=====	=====	
	TOTALS :	29,837	2,814	8.6%

Since the first PLN issue, the requests for translations have grown at a compound rate of 12% per month from 191 to 481. The growth in overall newsletter requests demonstrates that NCR University’s message is getting out and the growth in translation requests strongly suggests that the message is also *getting in*.

Recently the author conducted a survey of the recipients of the MT copies to determine more precisely how they view the quality and usefulness of the newsletter translation. The survey showed that 84% of the recipients found the translation “fairly useful,” “very useful,” or “essential” and of those about half were in the latter two categories. Sixty-four percent said that they would recommend the translation service to their colleagues and, more tellingly, 16% said they would not even read the newsletter were it not translated into their language. A significant increase over the proverbial ten percent who never get the word...

6 Lessons Learned

The project described in this paper took more than two and a half years and is still moving forward. Reflecting on our successes and failures, a few key observations and recommendations stand out – both because we did them and they worked and because we didn't and it hurt. Here is the list; brief explanations follow.

1. Start with a limited amount of content
2. Don't let internationalists speak for target users
3. Guide users' expectations and always give them a choice
4. Budget for ongoing maintenance, not just system purchase
5. Use rapid prototyping for a phased implementation
6. Develop a partnership relationship with the MT supplier
7. Persevere

NCR has an extensive global intranet. In our enthusiasm for bringing maximum value to our international colleagues it was tempting to propose a system that would provide instant translation services for all English content. This impulse is best resisted in favor of a more circumscribed corpus as, in our case, information about training programs offered through NCR University.

The target audience is *not* the internationalists. Every global corporation has people who are fluent in several languages and who appreciate and enjoy the nuances of transnational communication. Although they are not trained linguists, these are the people who are usually called upon to assist with internal translation projects and, in general, they will not be satisfied with the output of pure MT running against uncontrolled source text. The primary beneficiaries of this technology are those employees who are not fluent in the source language and who will be made more productive by using the translations. While this distinction seems obvious, the lesson we learned is that the vocal disdain of the internationalists we consulted at the beginning of the project almost drowned out the growing chorus of encouragement that we received from true end-users as the project progressed. Remember the words of Samuel Johnson, "Nothing will ever be accomplished if all objections must first be overcome."

Make it clear to users that they are using MT and always give them a choice to change their choice back to the source language. In addition to the MT advisory banners described in Section 3, we preface each of our translated issues of the Personal Learning News with these statements.

"Machine translation is a new technology that produces output that can be incorrect or hard to understand. If you speak English well you may prefer to *change your preference to English*. If you have difficulty with English, we think you may find this translation useful. If you have any questions about the translation, you can easily *view the original English version*. If there are any discrepancies, the English original takes precedence. "

The phrase “*change your preference to English*” is a hyperlink that takes users directly to a web page on which they can update their personal profiles. The phrase “*view the original English version*” is another hyperlink that opens a pop-up window containing the original English text.

Successful MT implementations require resources for continuous updates and improvement. The cost of these resources typically exceeds what is normally budgeted for software upgrades and maintenance as a percent of purchase price. Most MT systems have “tuning” files that specify preferred translations of selected words and phrases. They may also have lists of words that should not be translated due to their widespread use in their original forms. Many of the problems identified in our initial evaluation and many of the poor translations cited by our current users can be corrected using these control files.

The continuous maintenance of these files require people, in some cases linguists. The cooperation of rank and file users is also helpful to identify areas that need improvement. In our most recent survey, 71% of our MT users said they would be “willing to spend a few minutes each month to provide this input” – the identification of errors and suggested improvements for their languages. Our challenge now is to find ways to channel this volunteer energy into direct and constructive inputs to our control files.

Our experience suggests that big projects involving new technologies are amenable to implementation in stages using a series of rapid prototypes with ever-larger user groups. We also found that our time scales were longer than we anticipated and we were grateful for the fact that we had developed a long-term partnership relationship with our supplier (SYSTRAN). This allowed us to weather a number of setbacks and delays without jeopardizing the overall success of the project. (Given the technical nature of MT projects, it is also very desirable to have a good liaison person on the supplier’s technical staff – preferably one of the designers, not a help desk technician.)

Persevere.

An Assessment of Machine Translation for Vehicle Assembly Process Planning at Ford Motor Company

Nestor Rychtycky

Information Technology Services
Ford Motor Company
nrychtyc@ford.com

Abstract: For over ten years, Ford Vehicle Operations has utilized an Artificial Intelligence (AI) system to assist in the creation and maintenance of process build instructions for our vehicle assembly plants. This system, known as the Direct Labor Management System, utilizes a restricted subset of English called Standard Language as a tool for the writing of process build instructions for the North American plants. The expansion of DLMS beyond North America as part of the Global Study Process Allocation System (GSPAS) required us to develop a method to translate these build instructions from English to other languages. This Machine Translation process, developed in conjunction with SYSTRAN, has allowed us to develop a system to automatically translate vehicle assembly build instructions for our plants in Europe and South America.

1 Introduction

The Direct Labor Management System (DLMS) was developed at Ford Vehicle Operations as a tool to assist in vehicle assembly process planning. The major input to the DLMS system is a process sheet, which describes actual build instructions for a vehicle. The process sheets are written in a controlled language, known as Standard Language that was developed specifically for this application. The process sheets are read by the DLMS system and utilized to create detailed work tasks for each step of the assembly process. These work tasks are then released to the assembly plants where specific workers are allocated for each task. With the expansion of DLMS to our plants around the world, it became necessary to translate these instructions into the home languages of our assembly plants in Germany, Belgium, Spain, Mexico and Brazil. It was decided to utilize an automated translation solution due to the nature of the controlled language, amount of data that needed to be translated, and the frequency in which the data changes. A Machine Translation solution is more cost-effective than manually translating all of our process instructions and it could be used to translate process build instructions for other car lines into the target languages. In this paper we discuss our experiences with implementing machine translation for the DLMS application at Ford. We will include the following items: a discussion of the DLMS system in more detail, Standard Language, a discussion of our experiences with implementing language translation for DLMS and the architecture of our translation system. The paper concludes with a summary of our experiences with machine translation and describes future work.

2 The Direct Labor Management System

The Direct Labor Management System (DLMS) is an implemented system utilized by Ford Motor Company's Vehicle Operations division to manage the use of labor on the assembly lines throughout Ford's vehicle assembly plants. DLMS was designed to improve the assembly process planning activity at Ford by achieving standardization within the vehicle process build description and to provide a tool for accurately estimating the labor time required to perform the actual vehicle assembly. In addition, DLMS provides the framework for allocating the required work among various operators at the plant and builds a foundation for automated machine translation of the process descriptions into foreign languages.

The standard process-planning document known as a process sheet is the primary vehicle for conveying the assembly information from the initial process planning activity to the assembly plant. A process sheet contains the detailed instructions needed to build a portion of a vehicle. A single vehicle may require thousands of process sheets to describe its assembly. The process sheet is written by an engineer utilizing a restricted subset of English known as SLANG (Standard LANGUAGE). Standard Language allows an engineer to write clear and concise assembly instructions that are machine-readable. The DLMS system interprets these instructions and generates a list of detailed actions, known as allocatable elements that are required to implement these instructions at the assembly plant level. These allocatable elements are associated with MODAPTS (MODular Arrangement of Predetermined Time Standards) codes that are used to calculate the time required to perform these actions.

MODAPTS codes are utilized as a means of measuring the body movements that are required to perform a physical action and have been accepted as a valid work measurement system around the world [1]. DLMS is a powerful tool because it provides timely information about the amount of direct labor that is required to assemble each vehicle, as well as pointing out inefficiencies in the assembly process. A more complete description of the DLMS system can be found in [2].

3 Standard Language

Standard Language was developed as a standard format for writing process descriptions. Prior to the introduction of Standard Language, process sheets were written in free form text, which caused major problems because of ambiguity and lack of consistency. Standard Language was developed internally at Ford by systems people in conjunction with industrial and manufacturing engineers. The use of Standard Language has eliminated almost all ambiguity in process sheet instructions and has created a standard format for writing process sheets across the corporation.

Standard Language is a controlled language that provides for the expression of imperative English assembly instructions at any level of detail. All of the terms in Standard Language with their pertinent attributes are stored in the DLMS knowledge base in the form of a semantic network-based taxonomy. Verbs in the language are associated with specific assembly instructions and are modified by significant adverbs where appropriate. Information on tools and parts that are associated with each process sheet is used to provide extra detail and context.

The Standard Language sentence is written in the imperative form and must contain a verb phrase and a noun phrase that is used as the object of the verb. Any additional terms that increase the level of detail, such as adverbs, adjuncts and prepositional phrases are optional and may be included at the process writer's discretion. The primary driver of any sentence is the verb that describes the action that must be performed for this instruction. The number of Standard Language verbs is limited and each verb has been defined to describe a single particular action. The object of the verb phrase is usually a noun phrase that describes a particular part of the vehicle, tool or fastener. The process sheet writer may use prepositional phrases to add more detail to any sentence. Certain prepositions have specific meaning in Standard Language and will be interpreted in a predetermined manner when encountered in a sentence. Figure 1 shows how the Standard Language sentence "Feed 2 150 mm wire assemblies through hole in liftgate panel" is parsed into its constituent cases.

```
(S (VP (VERB FEED)) (NP (SIMPLE-NP (QUANTIFIER 2) (DIM
(QUANTIFIER 150) (DIM-UNIT-1 MM)) (ADJECTIVE WIRE) (NOUN
ASSEMBLY))) (S-PP (S-PREP THROUGH) (NP (SIMPLE-NP (NOUN
HOLE) (N-PP (N-PREP in) (NP (SIMPLE-NP (ADJECTIVE
LIFTGATE) (ADJECTIVE OUTER) (NOUN PANEL)))))))))
```

Figure 1: Sample of parse tree structure in DLMS

Any errors are flagged by the system and returned to the engineer to be fixed before this process sheet can be released to the assembly plant. The process engineers are also trained in the proper way to write sheets in Standard Language. Other facilities, such as a web-based help facility, training classes and support personnel are provided to assist the process engineer in the writing of correct process sheets in Standard Language. The vehicle assembly process is very dynamic; as new vehicles and assembly plants are added to the system, it requires that Standard Language also evolve. Changes to Standard Language are requested by process engineers and then approved by the Industrial Engineering organization; these changes are then added into the system by updating the DLMS knowledge base. The DLMS knowledge base and the AI system are maintained by the internal Ford systems organization.

The greatest improvements to machine translation quality can be made by limiting the expressiveness and grammar of the source text by imposing restrictions on that text. This restricted language improves the quality of translation by limiting both the syntax and vocabulary of the text that will be translated. These restrictions must be enforced through the use of a checker that will accept or reject text based on its compliance with those rules. Controlled languages have been utilized successfully in such diverse applications as weather forecasting, airplane manufacturing, heavy equipment manufacturing and automobile service technology [3].

4 Machine Translation in DLMS

The restricted grammar of Standard Language and its limited vocabulary made the DLMS application a very strong candidate for machine translation. Similar applications with controlled languages have successfully utilized machine translation with a very high accuracy rate [4]. After an evaluation of the commercial translation packages available at the time, we decided to work with SYSTRAN for our machine translation needs. This decision was based on the wide range of languages available, the high performance of the SYSTRAN translation software, the ability to run the software under the UNIX operating system and the willingness of SYSTRAN to customize their products for our application.

The use of machine translation for Standard Language provided us with many unique challenges. The amount of data we need to translate consists of about 50,000 records for each vehicle and each language pair. Currently, we are translating over 15 such vehicle language pairs. Each sentence of text is stored as a single record in an Oracle database, which has to be retrieved, translated and then written back into the database. Issues with performance and database integrity had to be addressed and an interface was developed by SYSTRAN and Ford to improve the translation performance.

Standard Language has adopted some grammatical structures not found in general English in order to optimize its ability to encode time and motion information. In addition, translation of standard language terms has required that similarly well-defined equivalents be designated in our target languages. Both of these have led to some specific problems in implementing machine translation.

Many of the structural decisions that were made in the development of Standard Language had a serious impact on the implementation of automated language translation technology for this application. The sentence structure in Standard Language is always imperative with the verb phrase in the beginning of the sentence. The verb phrase could also use a modifier that could impact the meaning of the verb. This modifier is in the form of an adverb, but in some cases, this word is a noun or an adjective that functions as an adverb. For example, the term “Robot Spot-weld the Object” uses the word “Robot” to modify the verb. These types of unconventional grammatical usage caused problems for SYSTRAN, which was developed to work with common English grammar.

A critical issue in our translation quality is in the use of part descriptions that are represented as long noun phrases: (ex. Side member bracket assembly medium). If the entire phrase is not present in our technical glossary, the system will translate the phrase by splitting it up or on a word-by-word basis. In most cases, this translation is incorrect and must be manually translated by our engineers at the assembly plants.

The Ford technical terminology raised the challenge of defining translation equivalents for the well-defined terms of standard language. There are many terms that describe automotive processes and parts that are utilized only within Ford Motor Company. These terms include acronyms, abbreviations, Ford locations and other terms that cannot be translated by anybody who is not familiar with Ford. It was found that many of the terms were not understood by all of our people, as they may only be used within one department in a plant. These terms all have to be identified and translated manually so they can be added into the SYSTRAN dictionaries

correctly. Problems were also caused by entries (ex: “shotgun”) that are used informally to describe tools or equipment at the plant. Many other people may be unaware of what this term represents and a literal translation of “shotgun” would make no sense to anybody in German or Spanish. Technical glossaries, such as those published by the Society of Automotive Engineers, are very useful in some cases, but they do not always contain a complete list of terms and can become dated and obsolete due to the rapid pace of technological progress.

Another issue with Standard Language deals with multiple spellings and misspellings of various terms. The DLMS system has a utility to allow engineers to add new terms to the system, as they are required. However, this also allows for multiple variations of the same concept due to misspellings or inconsistent usage. For example, some people would add acronyms without periods (ABS) and others would add the term with periods (A.B.S.). Over time, the knowledge base would contain quite a number of these variable spellings for the same object. Attempts were made periodically to clean up the knowledge base, but multiple terms did not become an issue until they all had to be translated.

As mentioned previously, the verbs in Standard Language are defined very concisely and unambiguously to represent a single particular action. It was not always possible to translate these verbs into other languages on a one-to-one basis to preserve their consistent meanings. The translation was accomplished only after spending considerable time on redefining their meanings in English and then translating the verb based on the most common usage in the target language. In some cases, one single English verb would have multiple translations based on its context or object that it was acting upon. Another problem arose with the use of compound verbs, which are a creation of Standard Language. A compound verb (ex: press-and-hold) was created to describe two actions with one verb that often occur together. Their usage makes it simpler for the process writers but causes complications in translations, as we are creating a new word in another language. The entire issue of defining an equivalent Standard Language lexicon for each of the target languages required considerable effort and is not entirely completed.

In Standard Language the use of articles is optional and they are usually not written in order to save time. This leads to sentence structures, which can be easily misinterpreted by the language translation software as it is expecting complete English sentences. This problem was partially solved by modifying the parser in the AI system to add articles into the text where appropriate. Another extension to Standard Language allowed for the usage of certain adjective modifiers after the noun in order to override some attribute of the part. This structure also caused problems during translation and the parser was modified to handle these problems also. Abbreviations are also expanded into their full size before the translation to prevent similar errors with their meaning.

Another problem involves the use of comments within Standard Language. Any comment or remark can be included into Standard Language if it is delimited from the regular text with brackets. These comments are ignored by the AI system and do not have to conform to the Standard Language rules. The translation of free-form text and individual words is extremely unreliable and continues to cause problems. Often the comments are valid terms that should be added into the Standard Language lexicon, but the process writer uses the commenting feature to bypass the process of adding these new terms into Standard Language.

Standard Language was never designed to produce regular grammatical sentences; the goal was to develop a consistent and understandable means of communicating engineering instructions. The initial implementation of Standard Language in our North American assembly plants also encountered user resistance until the process engineers were trained and learned how to use it effectively. It is not a surprise that the translation of Standard Language is also being resisted, as the user community needs to have an understanding of Standard Language before it can accept these translations. In this case the user community includes the plant personnel at the assembly plants that are building our vehicles. The machine translation system cannot be expected to produce exact grammatical translations in target languages and we have had some difficulties in getting this point across to our users.

5 Implementation of Machine Translation

The machine translation system was implemented into GSPAS through the development of an interface into the Oracle database. Our translation programs extract the data from an Oracle database, utilize the SYSTRAN system to complete the actual translation, and then write the data back out to the Oracle database.

Our user community is located globally. The translated text is displayed on the user's PC or workstation using a graphical user interface through the GSPAS system. The translation software was developed and modified to our specifications by SYSTRAN. It runs on Hewlett Packard workstations under the HP UNIX operating system. The Ford multi-targeted customized dictionary that contains Ford technical terminology was developed in conjunction with SYSTRAN and Ford based on input from human translators.

One of the most difficult issues in deploying any translation is the need to get consistent and accurate evaluation of the quality of your translations (both manual and machine). We are using the J2450 metric developed by the Society of Automotive Engineers (SAE) as a guide for our translation evaluators [5]. The J2450 metric was developed by an SAE committee consisting of representatives from the automobile industry and the translation community as a standard measurement that can be applied to grade the translation quality of automotive service information. This metric provides guidelines for evaluators to follow and describes a set of error categories, weight of the errors found and calculates a score for a given document. The metric does not attempt to grade style, but focuses primarily on the understandability of the translated text. The utilization of the SAE J2450 metric has given us a consistent and tangible method to evaluate translation quality and identify which areas require the most improvement.

We have also spent substantial effort in analyzing the source text in order to identify which terms are used most often in Standard Language so that we can concentrate our resources on those most common terms. This was accomplished by using the parser from our AI system to store parsed sentences as shown above in the database. Periodically, we run an analysis of our parsed sentences and create a table where our terminology is listed in use of frequency. This table is then compared to the technical glossary to ensure that the most-commonly used terms are being translated correctly. The frequency analysis also allows us to calculate the number of

terms that need to be translated correctly to meet a certain translation accuracy threshold.

A machine translation system, such as SYSTRAN's, translates sentence by sentence. A single term by itself cannot be translated accurately because the system does not know what part of speech it represents. Therefore, it is necessary to build sample test cases for each word or phrase that we will need to test for translation accuracy. This test case utilizes that term in its correct usage within the sentence. A file containing these translated sentences (known as a test corpus) is used as a baseline for regression testing of the translation dictionaries. After the dictionary is updated, the test corpus of sentences is retranslated and compared against the baseline. Any discrepancies are examined and a correction is made to either the baseline (if the new translation is correct) or the dictionary (if the new translation is incorrect).

The translation quality is evaluated both by SYSTRAN linguists and the users of the system. We have had difficulty in measuring our progress as opinions of translation quality vary significantly between translation evaluators. We have also allowed the users to manually override the translated text with their preferred translation. These manual translations are not modified by the system, but have to be redone each time that the process sheet is revised.

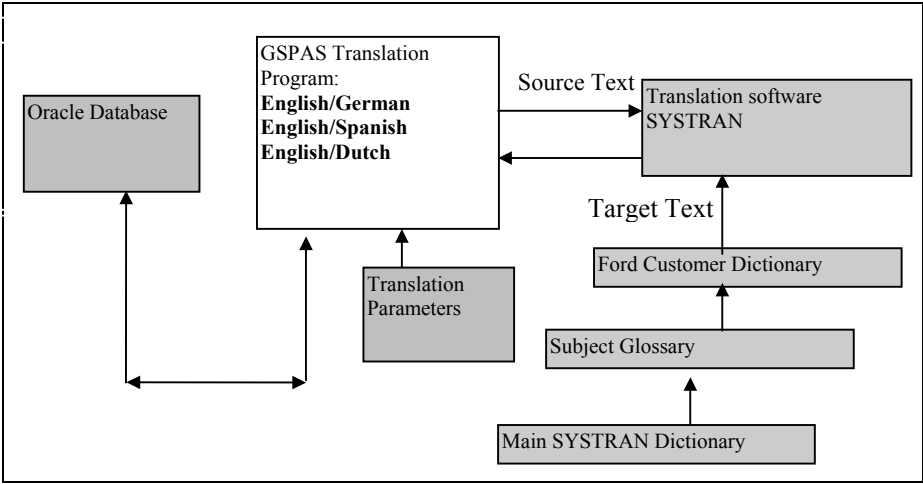


Figure 2: Machine Translation in GSPAS

6 On-Line Dictionary Management

We have recently implemented a web-based dictionary manager from SYSTRAN that allows us to update our technical glossaries in a timely fashion. The entire technical glossary is stored within an Excel spreadsheet that is maintained by Ford people who are proficient in both the technical terminology and the target language. These people can add or change a translation of a specific word or phrase in the spreadsheet and

submit the spreadsheet to SYSTRAN through a web-based interface. A new dictionary for each language is created and available for downloading within fifteen minutes. This new dictionary can then be immediately used by the machine translation system to produce the required results in the translation process. The linguistic coding required to modify the spreadsheet is quite minimal and has been extensively used by our engineering community. This rapid turnaround allows us to make changes to the technical glossaries and improve the translation quality in a timely fashion.

7 Conclusions and Future Work

Standard Language and DLMS has been in use at Ford for over ten years and has evolved from a prototype being tested at a single assembly plant to a fully-deployed application that is being utilized at Ford's assembly plants throughout the world, and has become an integral part of Ford's assembly process planning business. The use of machine translation in the GSPAS project allowed us to quickly translate large amounts of Standard Language text into the target language. To date, we have translated more than one million records in the four languages that we are currently processing.

The linguistic aspect of the Machine Translation has progressed much more slowly. As mentioned previously, the use of various unconventional structures in Standard Language had a very serious impact on the quality of the translations. The Standard Language dictionary was also open to user updates and resulted in the addition of errors and unnecessary terminology that caused errors in translation. To fix this problem, we have instituted the following steps: all additions to the dictionary are reviewed for correctness and applicability by a committee of representatives from various engineering organizations and we have instituted a process to clean up the dictionary and remove all unnecessary and incorrect terminology. We have very high hopes for the recently deployed web-based customer dictionary update tool. Our biggest problem continues to be the lack of bilingual engineers who understand the Ford technical language and can spend time evaluating and correcting the translation output. These engineers have many other responsibilities and it is very difficult to obtain their services for any length of time. We have also had some success in hiring retired Ford engineers for these kinds of tasks.

The evaluation and correction of translation text is often time-consuming as the participants live in Europe, Detroit and San Diego. The people at our European assembly plants, who are best suited for evaluating and correcting translated text, are also difficult to utilize due to their heavy workload. The user acceptance of machine translation also varies significantly; our goal is to deliver "understandable translations," but this means different things to different people. The current translation accuracy as informally measured by our user community is still unacceptable in some cases and we need to improve the performance of the system to gain user acceptance. We believe that we can improve the translation accuracy by making corrections to the dictionary, modifying the Standard Language text before translation and doing a better job of translating the non-Standard Language comments. It is important to note that in some ways, despite the restricted grammar, Standard Language is more difficult to translate than regular colloquial English. This is due to the specialized

terminology, the ungrammatical sentence structure and the style of the text. We have also given our users the ability to override the machine translation text when they feel that it is not understandable. The number of manual overrides varies from 0% to as high as 75% in cases (such as titles) where engineers frequently write incomplete sentences or phrases with many abbreviations and terms that are not supported in Standard Language. Our other plans include improving Standard Language training in Europe and incorporating linguistic support for the translation evaluation process in Europe. There is still considerable work to do to achieve our goal of 90% accuracy; nevertheless, we have made substantial progress with our machine translation application and plan to continue using this technology within Ford.

Acknowledgements

This project is a product of many people who have contributed to the system over the years; therefore, I would like to give credit to the following people: Rick Keller, Michael Rosen, Mike Boumansour, Randy Beckner, Jeff Miller, Juergen Koehler and a host of others who lent their assistance. The SYSTRAN team included Laurie Gerber, Jin Yang, Brian Avey, Carla Camargo-Diaz, Jean Senellart, Jean-Cedric Costa, Pierre-Yves Foucou, Christiane Panissod and Denis Gachot. I would especially like to thank Reba Rosenbluth who encouraged me to write this paper. Special thanks also go to the anonymous referees whose many comments and suggestions improved this paper immensely.

References

1. Industrial Engineering Services: Modapts Study Notes for Certified Practitioner Training (1988)
2. Rychtyckyj, N.: "DLMS: Ten Years of AI for Vehicle Assembly Process Planning," AAAI-99/IAAI-99 Proceedings, Orlando, FL, AAAI Press (1999) 821-828
3. Mitamura, T. & Nyberg, E.: Proceedings of the Second International Workshop on Controlled Language Applications, Carnegie Mellon University Language Technologies Institute (1998)
4. Isabelle, P.: "Machine Translation at the TAUM Group," in M. King (ed.), Machine Translation Today, Edinburgh University Press (1987) 247-277
5. Society of Automotive Engineers: J2450 Quality Metric for Language Translation, www.sae.org (2002)

Fluent Machines' EliMT System

Eli Abir, Steve Klein, David Miller, and Michael Steinbaum

Fluent Machines, 1450 Broadway, 40th Floor, New York, NY 10018
info@fluentmachines.com

Abstract. This paper presents a generalized description of the characteristics and implications of two processes that enable Fluent Machines' machine translation system, called EliMT (a term coined by Dr. Jamie Carbonell after the system's inventor, Eli Abir). These two processes are (1) an automated cross-language database builder and (2) an n-gram connector.

1 System Builder

Eli Abir (e-mail: eli@meaningfulmachines.com)

2 System Category

Research/Pre-Market Prototype

3 System Characteristics

EliMT is designed to enable MT from any language to any other language. The system is under development and has not been subject to full testing. Components of the two core processes have been tested independently under limited conditions for English–French (using a 60 million word corpus), English–Spanish (32 million word corpus), and English–Hebrew (2.5 million word corpus). These tests validated the two processes, however, performance metrics for the system's components (as well as for the integrated system) are not yet available. Full performance metrics for a variety of bi-directional language pairs (including English–French, English–Spanish, English–Hebrew, and English–Chinese) are anticipated to be available by October 2002.

4 System Description

Fluent Machines' technology is based on the concept that there exists an infinite number of sentences in any language, but a finite number of discreet ideas. These ideas, which Fluent Machines calls the "DNA" of meaning, are universal and can be expressed in any language. Furthermore, they can be linked together in any language to successfully express an unlimited number of complex ideas.

The Fluent Machines system breaks down source language input text into its DNA components, translates those components into the target language, and then accurately connects the translated components. At its core, the system is comprised of two processes: (i) the first process automatically builds large cross-language databases of basic DNA n-grams of various lengths (i.e., the system does not fix a length nor is it limited by length) and (ii) the second process accurately connects n-gram translations in the target language, thereby producing translated text.

Below is a description of the two processes and their implications. Because each process is proprietary, with patents pending, only a generalized treatment is provided.

4.1 Process 1: Automated Cross-Language Database Builder ("Database Builder")

Fluent Machines' first process enables the system to examine parallel text and automatically generate a database of n-gram translation pairs (regardless of length). Each of these n-grams is referred to in EliMT as a piece of DNA. This use of parallel text and statistical methods has some similarities to Statistical MT, however, the matching algorithm in EliMT has the ability to determine the translation of n-grams with statistical certainty.

Fluent Machines is also developing methods that are not dependent on parallel text to determine n-gram translations. These methods, called "AIMT," use large, independent sources of text in both the source and target languages (e.g., the Internet) in combination with existing translation methods or word-for-word dictionaries. AIMT requires more processing power but should offer a greater likelihood of successfully translating any given n-gram and thus providing much broader coverage of the language pairs being translated.

4.2 Process 2: Word-String Connection Process ("N-gram Connector")

Fluent Machines' second process connects contiguous n-grams in a target language only if the system knows with certainty that the connection will yield an accurate, longer word-string. This process was confirmed in tests using English–French, English–Spanish, and English–Hebrew. Until all the DNAs in a language pair are identified, incorrect connections are theoretically possible but anticipated to occur in extremely rare cases. Because there is a process that confirms the connection of each contiguous n-gram pair, the system constructs an n-gram chain that is locked into a coherent phrase, sentence or even longer word-string.

The N-gram Connector has several implications:

The ability to accurately combine DNA building blocks allows the system to build (and connect) n-grams from the limited universe of DNAs (at any given time) into an infinite number of sentences. The finite number of DNA building blocks needed to translate among languages provides EliMT with an identifiable path to closure.

Each translation added to the database increases the ability of the system to accurately translate text by a large multiple as all naturally connecting n-grams in the database can combine with the new database entry.

EliMT can specifically identify potentially incorrect portions of its translation with certainty.

4.3 Accuracy vs. Completeness

There is an important distinction between accuracy and completeness. The Database Builder is responsible for completeness, while the N-gram Connector is responsible for human-quality accuracy. Until a cross-language database is complete, the system will yield human-quality, but not complete, translations. The system's ability to identify with certainty which portions of the text have been translated correctly will be a valuable aid to human translators. As the cross-language databases grow, the percentage of a document that is translated with human-quality accuracy will increase.

4.4 Every Language to Every Language

The Database Builder can analyze parallel text between any two languages. As each language pair is added, the system should build database entries faster and more efficiently by leveraging known translation relationships among the languages it has previously analyzed.

Leveraging multiple languages has two important implications:

Adding text in any language pair increases the coverage for many other languages at the same time.

EliMT, like Statistical MT, builds engines between languages automatically, bypassing the costly and time-consuming development of rule-based and transfer-based systems. Moreover, the automatic building of databases *among* languages should provide a large number of translation engines. 9,900 different systems are needed to translate among the 100 most popular languages. It is anticipated that EliMT will be able to offer MT systems among all languages that have sufficient quantities of parallel text.

Historically, MT technologies have not achieved high levels of accuracy for language pairs that are very different from one another (e.g., English–Chinese). EliMT's approach, based on the "DNA" of meaning and the organic locks between n-grams, is extremely flexible and therefore well-suited for translation among very different languages.

4.5 The Critical Steps Remaining

Once the Database Builder analyzes a sufficient number of cross-language documents to build a database to critical mass, the N-gram Connector can combine them into accurate sentences. The number of DNA building blocks constituting critical mass and the amount of parallel text necessary to identify them is unknown at this time. We estimate critical mass for database entries to be between 1 billion and 5 billion entries per language.

If Fluent Machines is successful in its efforts to develop AIMT methods, the need for parallel text will be decreased (if not eliminated). However, for system efficiencies, the use of parallel text will, when available, be preferred.

5 More Information

More information on Fluent Machines can be found at our website:
www.fluentmachines.com.

LogoMedia TRANSLATE™, Version 2.0

Glenn A. Akers

Language Engineering Company, LLC and LogoMedia Corporation,
385 Concord Avenue, Belmont, Massachusetts, USA
glenn@lec.com
www.logomedia.net

Abstract. LogoMedia Corporation offers a new multilingual machine translation system – LogoMedia Translate – based upon smaller applications, called "applets", designed to perform a small group of related tasks and to provide services to other applets. Working together, applets provide comprehensive solutions that are more effective, easier to implement, and less costly to maintain. Version 2, released in 2002, provides a single set of cooperating user interfaces and translation engines from 6 vendors for English ⇔ Chinese (Simplified and Traditional) Japanese, Korean, French, Italian, German, Spanish, Portuguese, Russian, Polish, and Ukrainian.

LogoMedia Corporation introduced in 2001 a new family of translation products based upon smaller applications, called "applets" designed to perform a small group of related tasks and to provide services to other applets. The key to making such a system work is cooperation between the applets. Microsoft supports this communication through the Component Object Model (COM). With this technology, solutions can be written as a set of cooperating components, each using COM to provide or use services.

To make this vision of flexible, extensible machine translation a reality, Language Engineering Company developed the LEC Translation Model. This model is a set of COM interfaces that define how various translation components interact.

Following the LEC Translation Model, developers use these interfaces to build translation clients and translation engines, which can interact with clients and engines from other vendors. This technology is not intended just for LogoMedia products. Rather, the intent is to establish a standard that will promote interaction among all vendors of products that provide or use translation services. The result will be better solutions for the customers and a larger market for developers.

User Interface (UI) components should not need to be aware of the natural languages that will be translated. Products that can work with any language pair have wider appeal than applications that directly incorporate a particular translation technology. This ability to sell translation tools in a variety of markets will encourage developers to create higher quality tools that can work with translation systems that support these APIs.

The cost of developing natural language neutral translation clients is much less than the cost of developing separate clients for each language pair. With one client for all language pairs, it becomes viable to develop products for markets that were previously impenetrable, perhaps by bundling the client with translation engines developed by other companies.

Our system is based on MT engines and user interfaces used by more than a 2 million customers worldwide. Version 2.0 includes engines for translation between English ⇔ Japanese, Chinese, Korean, English, French, German, Italian, Polish, Portuguese, Russian, Spanish, and Ukrainian. We have combined our UI applets with the best available translation engines from six leading translation engine developers, to provide a unified interface for 22 translation pairs in version 2.0. We will soon add 14 additional pairs from some of the same developers: German ⇔ French, Italian, Russian, Ukrainian; Italian ⇔ French, Spanish; French ⇔ Spanish. With 36 high quality direct translation pairs, our offerings are extensive, and we will soon offer additional pairs through our own development (English ⇔ Turkish, Dutch) and through licensing (English, French ⇔ Arabic).

TRANSLATE consists of many components. Each component provides a unique service.

- *Translation clients* provide user interfaces for translation services in particular contexts.
 - **LogoTrans** is useful for fast and easy translation of small amounts of text. LogoTrans translates text in three ways: as you type, when you drag and drop text from another application, or by watching for changes on the clipboard. Source text appears in the upper LogoTrans pane. Translations appear in the lower pane. No commands are needed - the translation happens automatically and instantly.
 - **FileTrans** provides batch translation of large amounts of text and HTML files, or even folders containing text and HTML files.
 - **TransIt** is designed to work with all applications that accept text entry. For example, you can use TransIt to specify search text in an Internet browser, or to reply to email. The TransIt window has an area on the right for inputting text. You simply type or paste text in, hit <Enter> to have it translated, and then hit <Enter> again to put the translation into your front most application window at the cursor.
 - **Translation Mirror** translates the contents of a different application's window, automatically responding to changes made to the text in that window. No commands are needed - the translation happens automatically and instantly.
 - **MS Word add-in** adds a LogoMedia menu to Word to allow translation by LogoMedia engines from within Word.
- Individual *Translation Engines*, including language-specific dictionaries and options, do the actual computations to translate text strings for each language pair. All of the other components use the translation engines when translations are needed.

- *Translation Options Editors* provide user interfaces to set the preferences for translation. The translation clients may use a shared version of the preferences for a particular language pair, or they may manipulate and store individual preferences.
- *Dictionaries* display or manipulate the contents of system, technical, and user dictionaries. Dictionaries provide the dictionary editing functions for any language pair when needed.
- The *Options Control Panel* is a user interface applet to control the interface language, fonts, and translation engine versions used by all of the translation tools. Current user interface languages are English, Japanese, and Spanish.
- The *Tip of the Day* applet displays hints about the various applets to the user.

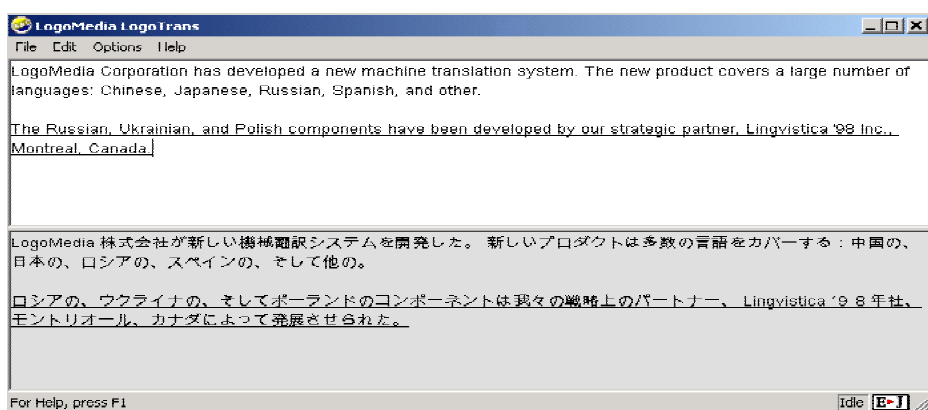


Fig. 1. shows direct English to Japanese translation using LogoTrans™.

The Composite Translation Engine combines pairs to provide translations of those language pairs for which no direct translation engine exists. For example, a Ukrainian text can be translated into Chinese using English as an “interlingua”. First our system translates Ukrainian to English and then it translates the English output to Chinese. See Fig. 2 for an illustration of the user’s screen when translating Ukrainian text to Chinese.

To increase translation quality in specialized fields, many Technical Dictionaries are available, although not all of them for each translation pair. LogoMedia technical dictionaries contain over 4.4 million terms for Accounting, Aerospace, Agriculture, Architecture, Automotive, Aviation, Banking, Biology, Biotechnology, Business, Chemical, Chemical Engineering, Civil Engineering, Computer, Earth Science, Ecology, Economy, Electrical Engineering, Energy, Finance, Geology, Legal, Marine Technology, Materials Science, Mathematics, Mechanical Engineering, Medical, Metal Engineering, Microelectronics, Military, Oil & Gas, Patents, Physics, Sociology, Space Exploration, Telecommunications, Urban Engineering, and Zoology.

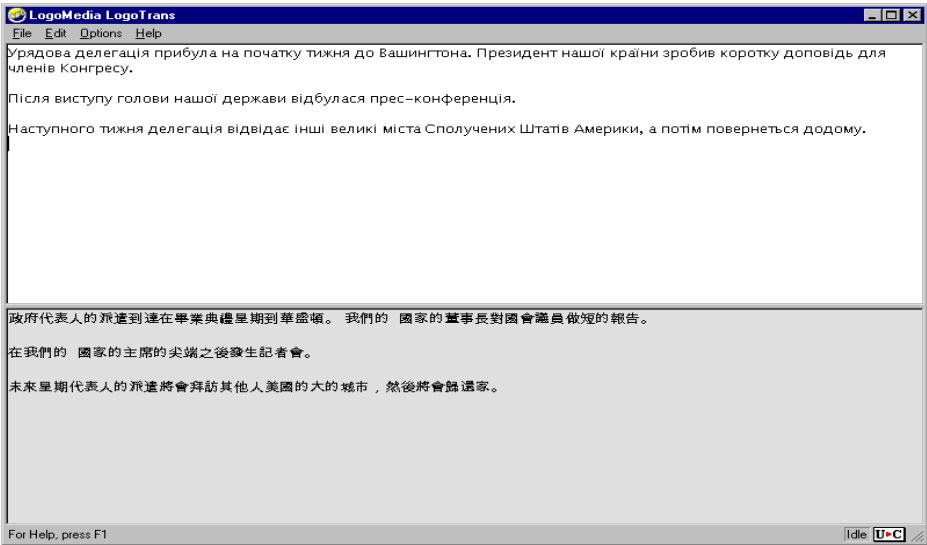


Fig. 2. Translating Ukrainian to Chinese using the Composite Engine

Translation Mirror™ has several uses. It can be used to monitor a web browser and provide immediate translations of each page. It is also used to automatically provide a back translation when composing messages or texts.

The LEC Translation Model defines a new way for machine translation components to interact. It decouples the different aspects of translation, allowing each component to be built more simply and easily. Translation clients, translation servers, translation engines, and translation utilities can all cooperate while retaining their independence. Multiple translation language pairs and multiple uses for translations become opportunities rather than problems.

Another useful UI is the Microsoft® Excel Add-In which adds translation menu options to Microsoft Excel to allow translation of the contents of any series of cells.

For most users, the preferred operating systems for Translate are Windows® 2000 or XP because of their superior support for multiple languages and their stability. Translate also supports Windows ME, 98, ME, and NT4.0 (SP4 SP6 or later). At least 64MB of total system RAM, and 100MB to 1GB of disk space are required, depending upon the number of language pairs installed.

LogoMedia also provides translations via the internet (www.logomedia.net) using the same architectural philosophy and many of the same mechanisms described in this overview. However, our next generation implementation of internet translation will replace DCOM as its wire protocol with SOAP (Simple Object Access Protocol). We aim to release our initial SOAP version within 2002 with at least the 36 language pairs which now conform to LEC's Translation Model APIs.

Natural Intelligence in a Machine Translation System

Howard J. Bender

Any-Language Communications Inc.
4200 Sheridan Street
Hyattsville, MD 20782 USA
hjbender@any-language.net
202-413-6101

Abstract. Any-Language Communications has developed a novel semantics-oriented pre-market prototype system, based on the Theory of Universal Grammar, that uses the innate relationships of the words in a sensible sentence (the natural intelligence) to determine the true contextual meaning of all the words. The system is built on a class/category structure of language concepts and includes a weighted inheritance system, a number language word conversion, and a tailored genetic algorithm to select the best of the possible word meanings. By incorporating all of the language information within the dictionaries, the same semantic processing code is used to interpret any language. This approach is suitable for machine translation (MT), sophisticated text mining, and artificial intelligence applications. An MT system has been tested with English, French, German, Hindi, and Russian. Sentences for each of those languages have been successfully interpreted and proper translations generated.

1 The Approach

Current approaches to commercial MT systems have focused on statistical methods (such as IBM's research for the past 15 years [1]), example-based processing (such as Microsoft's research with their MindNet database [2]), or some combination (such as the Fluent Systems method [3]). Any-Language Communications has chosen a semantics-oriented approach with two syntactic components to develop a natural language understanding (NLU) and machine translation system. Rather than rely completely on artificial intelligence to interpret language meaning, our system exploits the natural intelligence embedded in a sentence by its human creator. This natural intelligence is extracted by relationship analysis software and is used to create equivalent expressions in any language. In addition, all language information is contained in the databases, permitting the code to be completely language-independent. The general process can be illustrated as follows:

SYNTAX ANALYZER ✂ SEMANTIC INTERPRETER ✂ WORD ARRANGER

Both NLU and MT proceed on sentence boundaries. The Syntax Analyzer identifies and parses source language sentences, identifies independent and dependent phrases, determines possible parts-of-speech, handles language-specific features (such as

possessives in English), determines the “subject” of each independent part, and links words that depend on each other (such as adjectives with their corresponding nouns).

The Word Arranger takes the results of the Semantic Interpreter and creates grammatically correct sentences in the target language. Words are extracted from the target language dictionary according to directions given (via codes) by the Semantic Interpreter, and those words are arranged using target language rules. Word linking information from the Syntax Analyzer is used to select proper word options and word endings, as the target language requires.

The Semantic Interpreter, the processing heart of the system, is based on five concepts: The Theory of Universal Grammar [4], a weighted inheritance system, a specially developed number language, a type of genetic algorithm, and incorporation of all language information in the database rather than in the code. By merging these concepts, the Semantic Interpreter selects the contextual meaning of each word or phrase in a sentence. The output of the Semantic Interpreter, a collection of codes used by the Word Arranger to construct target language sentences, also provides the true contextual meaning of each word, idiom, and word phrase selected. These codes can also be used by artificial intelligence applications (such as robotics or language learning) to form responses to users.

Semantic interpretation has been the major stumbling block to effective language-independent, unrestricted vocabulary NLU and MT. Consequently, the remainder of this paper will focus on how we have resolved this problem. The following sections explain how the concepts we use perform the semantic analysis.

1.1 Theory of Universal Grammar

This theory posits that language is composed of two components: A surface structure and a deep structure. The surface structure consists of the word order, word endings, parts of speech, etc. (the syntax), and is specific for each language. So, English has its own syntax, French has its own, and so forth. The deep structure is the actual meaning of the words (the semantics), and is universal across all languages. This means that the deep structure interpreted in English will have the same deep structure in French or in Chinese, or in any language. The Semantic Interpreter works with this deep structure.

1.2 Weighted Inheritance System

To implement deep structure analysis, we’ve developed a copyrighted class/category hierarchy. Containing five levels, in which each lower level inherits characteristics from the level above it, this hierarchy permits recognizing relationships between words. There are 16 classes and over 800 categories that include all language concepts. Classes include Living Things, Human Society, Behavior & Ethics, etc. Categories under Living Things include Sleeping, Eating, Medicine, etc. The weights assigned to each word within the class/category structure vary depending on relationships with other words in the sentence. In addition, we’ve found that language processing occasionally requires a third dimension of analysis to link category relationships beyond those found in the hierarchy. A language in each of the “major”

language families (comprising the first language of over 70% of the world's people [5]) has been inspected, and all languages contain every one of the categories. Those language families are the Chinese family, the Germanic family, the Indic family, the Japanese family, the Malayo-Polynesian family, the Romance family, the Semitic family, and the Slavic family. The system has been tested with English as the source language and French, German, Hindi, and Russian as the target languages.

1.3 Number Language

Words have always been difficult for computers to evaluate. Consequently, each word, idiom, and word phrase entered in the system is transformed into a number that represents its relative place in the class/category organization. By forming pairs of the sentence words/idioms/phrases and comparing the values of their relative places (adjusted by the class/category weights), a value for the pair relationship can be obtained. Such valuations are calculated for all pairs in the sentence.

1.4 Genetic Algorithm

The possible meanings for words quickly produce a massive number of possible sentence interpretations. Even a seven-word sentence can easily result in over 100,000 possible interpretations. While only a few of the sentences will be "sensible", the computer has no way of knowing which are and which aren't, and the combinatorial explosion of the analysis can overrun the processing capabilities of most computers. In fact, this was one of the major reasons for failure in early translation attempts. Recently, a mathematical technique called "genetic algorithms" was developed to address these "hard" problems, and has been applied to weather forecasting, pipeline analysis, traveling salesman problems, etc. Conceptually similar to the way body cells produce DNA, the most viable products survive to combine with other viable products to produce the "fittest" final product. In the Semantic Interpreter, partial sentence solutions are compared with each other, with the "best" ones remaining while the others are cast off. Through multiple combinations and adjustments, the best sentence is developed. This may be the first use of genetic algorithm methods for natural language analysis.

1.5 Database Design

One of the key features of the Any-Language Communications system is that the same Semantic Interpreter can be used to understand any natural language. This is possible because the information necessary to understand a language is contained in the language dictionary databases, not in the code. All that the Semantic Interpreter requires to process a particular language is to point to the dictionary for that language. Another feature of our database design is that words and idioms from each language point to their equivalent in all other languages in the system. The effect is that equivalent words and phrases are chosen among the languages, making high-quality contextual machine translation possible and also permitting simultaneous translation in any number of languages.

2 Language Interpretation

The Any-Language Communications system does a semantic analysis of every word in a sentence. That means there is no need for a keyword search and the vocabulary is unrestricted, permitting sentences on any subject to be processed. The system can recognize nuances not possible in systems that do partial analyses. For example, "The hot dog is ready to eat" and "The hot dog is ready to pant" are understood to mean a food and an overheated canine, respectively. This example also shows that the system understands idioms (some systems require "hot dog" to be written as "hotdog" when referred to as a food, and some always interpret "hot dog" as a food) and can understand what is meant even if the sentence is not grammatically correct (the first sentence should have been "The hot dog is ready to be eaten.").

The following examples show machine translations for some linguistically-interesting sentences, with English as the source language and French, German, Hindi, and Russian as target languages. Only very basic Syntax Analyzer and Word Arranger programs were used to support the Semantic Interpreter.

The hot dog is ready to eat.
 Le hot-dog est prêt pour manger.
 Die Frankfurter Wurst ist fertig zu essen.
 Yeh hot dog tayyar ke liye khaane hai.
 Сосиска готовая к еде.

The hot dog is ready to pant.
 Le chien chaud est prêt pour haleter.
 Der heisse Hund ist bereit zu hecheln.
 Yeh garam kutta razi ke liye hafne hai.
 Разгоряченная собака готовый к быстро дышит.

My refrigerator is running and my nose is running.
 Mon réfrigérateur fonctionnel et mon nez coule.
 Mein Kühlschrank läuft und meine Nase rinnt.
 Mera fridge chal raha hai aur mera naak beh raha hai.
 Мой холодильник бежать и мой нос течет.

My candidate is running.
 Mon candidat se présente aux élections.
 Mein Kandidat stellt sich der Wahl.
 Mera ummeedvar chunaav lad raha hai.
 Мой кандидат баллотируется.

My candidate is running a temperature.
 Mon candidat fait une fièvre.
 Mein Kandidat hat ein Fieber.
 Mera ummeedvar taral ko taap hai.
 ! " # \$ % & ' (' %) * + \$ %) * , - * . %) / . % .

The Any-Language Communications system uses its Semantic Interpreter independently of the target language. The fact that the database contains all the language information eliminates the need to have multiple semantic components. In most MT systems, each language pair is distinct, which means for n languages you would need $n(n-1)$ translation programs to service all of them. For example, to translate between 10 languages, you would need $10(10-1) = 90$ translation programs. Since the Any-Language Communications system uses the same Semantic Interpreter for all languages and requires only n Syntax Analyzers and n Word Arrangers for n languages, our example would need only 10 translation programs.

3 System Specifications

The Semantic Interpreter is written in Prolog. The Syntax Analyzer and Word Arranger programs can be written in any language that can interface with an EXE. The software is less than 500K bytes in size (excluding vocabulary databases) and can run on a standard PC, on a network, on the Web, or be embedded in a hardware product. The vocabulary databases will initially contain approximately 350,000 words and phrases in each language and be approximately 20 MB per language. Currently, the English database contains over 125,000 words and phrases, and we have seen no problems with scalability. Entries for other languages have been included for system testing. Our approach has no concept of “language pairs” - when the Syntax Analyzer, Word Arranger, and vocabulary databases are completed for a language, that language can be either the source or the target language for any other language in the system.

While the Semantic Interpreter has been completed, full system functionality requires Syntax Analyzer and Word Arranger software and vocabulary databases for each language to be translated. Any-Language Communications is continuing this development.

References

1. Brown, P., Della Pietra, S., Della Pietra, V., Mercer, R.: The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19 (1993) 263-311
2. Richardson, S., Dolan, W., Menezes, A., Pinkham, J.: Achieving Commercial-Quality Translation with Example-Based Methods. In: *Proceedings of the VIIIth MT Summit*, Santiago de Compostela, Spain (2001) 293-298
3. Technologic. http://www.talaris.com/assets/news_pdf/cp.pdf Computer Letter. March 11 (2002)
4. Chomsky, N.: *Syntactic Structures*. Mouton, The Hague (1957)
5. Hoffman, M. S. (ed.): *The World Almanac and Book of Facts*, 1990. Pharo Books, New York (1990)

Translation by the Numbers: Language Weaver

Bryce Benjamin¹, Kevin Knight², and Daniel Marcu²

¹C. Bryce Benjamin, CEO
Language Weaver, Inc.
1639 11th St., Suite 100A
Santa Monica, CA 90404
cbryceb@languageweaver.com

²Kevin Knight, Daniel Marcu
USC/ISI
4676 Admiralty Way
Marina del Rey CA 90292
{knight,marcu}@isi.edu

1 System Category

Pre-market prototype – to be available commercially in the second or third quarter of 2003.

2 Hardware Platform and Operating System

UNIX, Linux, Windows

3 Background

All automatic machine translation systems currently available on the market employ manually written rules that require many person-years to develop. These rules endeavor to spell out translations for all words and phrases. The rules also describe the restructuring of sentences for the target language, and the reordering of words and phrases to produce grammatical sentences. Extending and adapting such systems to work with different types of text, different topics, and new language pairs is extremely difficult, time-consuming, and error-prone. Such systems tend to reach a performance ceiling at around 70%-80% accuracy for open-domain text (using a word-based metric), after which any modifications to the system actually degrade performance, as the complex rules compete and interfere with each other.

Over a period of several years, an ISI/USC research team led by Dr. Kevin Knight and Dr. Daniel Marcu has developed a new, statistical/cryptographic approach to the automatic translation of human languages. The technology developed at ISI/USC was exclusively licensed to and is being productized and improved by Language Weaver Inc. In contrast to current commercial systems, the Language Weaver statistical

translation system uses statistical/cryptographic techniques that *automatically learn* how texts can be translated from one language into another. All that a statistics-based translation engine needs in order to handle new language pairs or types of text is a large collection of sentence pairs that are mutual translations of each other. The Language Weaver software takes an aligned, bilingual corpus of sentence translation pairs as input, and automatically produces a translation system capable of translating text using language and jargon similar to that of the corpus used for training.

In contrast to rule-based approaches, Language Weaver's development cost for new language pairs is very low. Also, the statistical translation technology developed by Language Weaver can be integrated into working environments with computer-aided translation tools to create capabilities and interaction modes that are beyond the current state of the art in computer-assisted translation technology. These capabilities, which may be applied in any translation environment, include access to automatically built dictionaries of words and phrases, the possibility to choose between multiple translations of the same sentence, and the possibility to have the system learn language specific to each domain and genre of interest.

4 System Characteristics

Language Weaver is the first commercially available, fully statistical machine translation system. Unlike other commercially available MT systems, Language Weaver does not have a hand-crafted rulebase, or a conventional dictionary. Language Weaver automatically extracts linguistic knowledge from bilingual text corpora. This knowledge includes word and phrase translations, source language re-ordering patterns, and target language grammar. All of this knowledge is applied simultaneously when we translate new, previously-unseen documents.

Because the knowledge is learned automatically, the system can be applied rapidly to new language pairs and new domains. Customization is done by retraining the system on parallel text in the desired topic and style. Whereas most commercial MT systems will allow users to customize the dictionary, and may allow users to write ad hoc translation rules. Language Weaver automatically learns the vocabulary of the text it is trained on, and even learns the style of the text it is trained on. Product functionality will include the ability to extend the delivered system vocabulary via list import, or via an "adapter" feature, which will learn new vocabulary and style parameters from customer corpora such as translation memory databases or other parallel texts.

Periodic evaluations of Language Weaver's technology show that the quality of its translations have increased over the last few years at a steady rate. When the Language Weaver French-English system was evaluated in 2001 on test data from the Canadian Hansards (the same corpus it was trained on), 41% of the sentences it translated were of publication quality, with no postediting. 28% of the sentences in the same test set were of publishable quality when translated by Systran. Presumably Systran could be customized to improve its performance as well, but dictionary building is laborious work, and it would certainly not be feasible to adapt Systran to the 1 million sentence corpus which Language Weaver learned automatically. While

41% accuracy may sound surprisingly low, coming from a developer, we are using an exceptionally conservative standard. MT systems are usually evaluated to approximate the number of *words* translated correctly, a much more forgiving standard. To test the robustness of our algorithms, we have also trained and tested translation systems for Chinese and Tamil to English.

The Language Weaver translation system is based on research and software done at the Information Sciences Institute (ISI) of the University of Southern California.

References

1. Al-Onaizan, Y., Knight, K.: Named-Entity Translation. In: Proceedings of ACL-02 (2002)
2. Yamada, K., Knight, K.: A Decoder for Syntax-based Statistical MT. In: Proceedings of ACL-02 (2002)
3. Hermann, U., Jahr, M., Knight, K., Marcu, D., Yamada, K.: Fast Decoding and Optimal Decoding for Machine Translation. In: Proceedings of ACL-01, Best Paper award (2001)
4. Marcu, D.: Towards a Unified Approach to Memory- and Statistical-Based Machine Translation. In: Proceedings of ACL-01 (2001)
5. Yamada, K., Knight, K.: A Syntax-Based Statistical Translation Model. In: Proceedings of ACL-02 (2002)
6. Hermann, U.: Building a Statistical Machine Translation System from Scratch: How Much Bang Can We Expect for the Buck? In: Proceedings of the Data-Driven MT Workshop of ACL-01. (2001)
7. Koehn, P., Knight, K.: Knowledge Sources for Word-Level Translation Models. In: Proceedings of Empirical Methods in Natural Language Processing (EMNLP'01) (2001)
8. Al-Onaizan, Y., Hermann, U., Hermjakob, U., Knight, K., Koehn, P., Marcu, D., Yamada, K.: Translating with Scarce Resources. In: Proceedings of the American Association for Artificial Intelligence conference (AAAI'00) (2000)
9. Koehn, P., Knight, K.: Estimating Word Translation Probabilities from Unrelated Monolingual Corpora Using the EM Algorithm. In: Proceedings of the American Association for Artificial Intelligence conference (AAAI'00) (2000)
10. Knight, K.: Decoding Complexity in Word-Replacement Translation Models. *Computational Linguistics*, 25(4) (1999)
11. Knight, K.: A Statistical MT Tutorial Workbook. Ms, August (1999)
12. Knight, K., K. Yamada.: A Computational Approach to Deciphering Unknown Scripts. In: Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing (1999)
13. Knight, K., Al-Onaizan, Y.: Translation with Finite-State Devices. In: Proceedings of the 4th AMTA Conference (1998)
14. Stalls, B., Knight, K.: Translating Names and Technical Terms in Arabic Text. COLING/ACL Workshop on Computational Approaches to Semitic Languages (1998)
15. Knight, K.: Automating Knowledge Acquisition for Machine Translation. *AI Magazine*, 18(4) (1997)
16. Knight, K., J. Graehl.: Machine Transliteration. In: Proceedings of ACL-97 (1997)

A New Family of the PARS Translation Systems

Michael Blekhman, Andrei Kursin, and Alla Rakova

Lingvistica '98 Inc. 109-11115 Cavendish Blvd., Montreal, QC, H4R 2M9, Canada
ling98@canada.com
www.ling98.com; www.lingvistica.com

Abstract. This paper presents a description of the well-known family of machine translation systems, PARS. PARS was developed in the USSR as long ago as in 1989, and, since then, it has passed a difficult way from a mainframe-based, somewhat bulky system to a modern PC-oriented product. At the same time, we understand but well that, as any machine translation software, PARS is not artificial intelligence, and it is only capable of generating what is called “draft translation”. It is certainly useful, but can by no means be considered a kind of substitution for a human translator whenever high-quality translation is required.

Introduction

PARS covers the following language pairs:

- English↔Russian – PARS 5.x
- English↔Ukrainian – PARS/U 3.x
- English↔Polish 1.x
- German↔Russian – PARS/D 6.x and PARS/D Internet
- German↔Ukrainian 4.x

All of them are fully compatible with each other and can be installed on the same computer.

Besides, PARS English↔Russian, English↔Ukrainian, and English↔Polish engines are used in the LogoMedia Translate system.

The main features of the new PARS family are as follows:

- they are compatible with all MS Windows versions including Windows XP;
- they are integrated with MS Word versions beginning from Word 97;
- they are integrated with MS Internet Explorer 5.5 and above;
- they ensure the drag-and-drop translation mode.

Besides, an Internet mining program has been developed for adding new words to the PARS dictionaries.

PARS can translate texts in the following modes:

- directly in MS Word – provided that the customer has MS Word 97, 2000, or XP installed on his/her computer;
- directly in Internet Explorer – provided that the customer has Internet Explorer 5.5 or above installed;
- in the drag-and-drop mode;
- besides, the customer can translate a file or the contents of the Clipboard.

Translating in MS Word

After the user run his/her MS Word 97, 2000, or XP, they can see that its Toolbar has changed, namely:

- a. There is the *Translate* command on the Toolbar, between *Tools* and *Table*. Clicking this command and then the translation direction, such as, for example, *Russian-English*, will translate the text opened in MS Word. The translation will appear under the source text, in a separate Word window.
- b. One can also see a box on the Toolbar displaying the current translation direction. One key click will translate the text opened in MS Word. The translation will appear under the source text, in a separate Word window.
- c. There are also several PARS text editing commands on the Toolbar:

- *Delete Variants* – deletes translation variants in the resulting text.
- *Options* – opens the Options dialog box, which lets one tune up one's PARS if necessary.
- *Desktop* – lets the user arrange the source text and its translation as two MS Word windows, one under the other.
- *New Word* – will add a word or phrase to the currently open dictionary: to do this, one should select the phrase or click on the word, and then click the *New Word* button.
- There are several buttons for linguistic text editing: changing the case (small/big letters), deleting a word, transposing two adjacent words, inserting articles and preposition *of*.

PARS Options

PARS provides a number of user-oriented options. Here are the most frequently used ones.

The *General* section lets the customer check/uncheck displaying translation variants. If this option is ON, each of the words having multiple translations will be marked with an asterisk in the resulting text. Double click an asterisk displays translation variants; a click on a more appropriate translation will paste it into the text instead of the previously displayed one.

English-Russian, *English-Ukrainian*, etc. display the list of dictionaries to be used in the translation session.

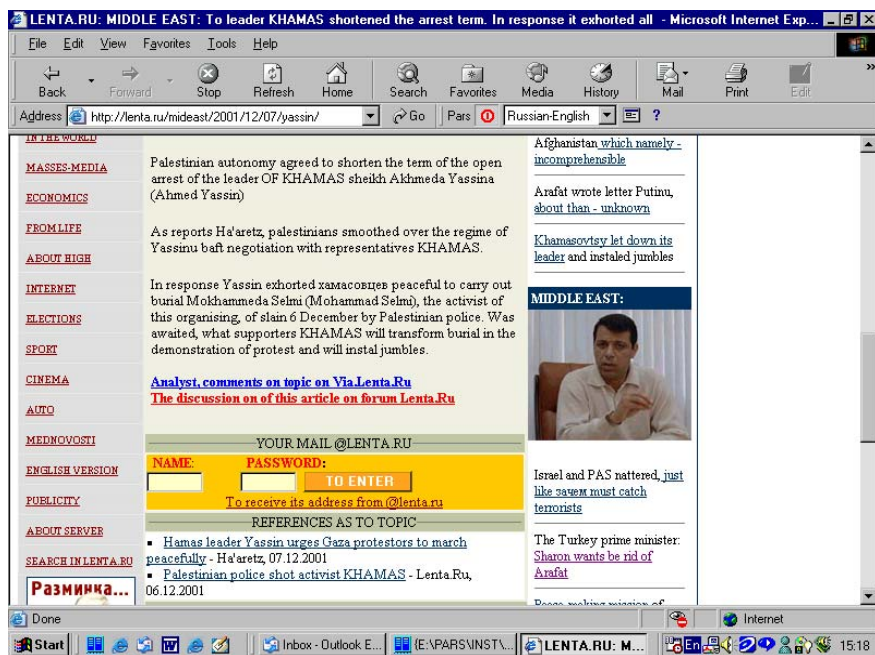
- PARS can use up to 4 dictionaries in each translation session. The user can remove a dictionary from the list, add a dictionary, or change dictionary priority.
- To add a dictionary, the user clicks *Add* and selects a dictionary name in the window that pops up.

Translating in the Internet Explorer

The user runs his/her Internet Explorer 5.5 or higher and sees the *Pars* icon on the Toolbar.

To have a web page translated, the user has to do the following:

- open a web page as he/she usually does in the Internet Explorer;
- select the translation direction, for example, Russian-English;
- activate PARS – either by pressing the red circle *Pars* button on the Toolbar, or by turning the translation on in *Tools\Options*;
- press the F5 button on the keyboard: the web page will be translated and displayed, the format will be fully preserved:



If one now clicks a link to open another page, such as (see the above screenshot) *The Turkey prime minister: Sharon wants be rid of Arafat*, translation will be made “on the fly”, and the corresponding web page will be opened and translated.

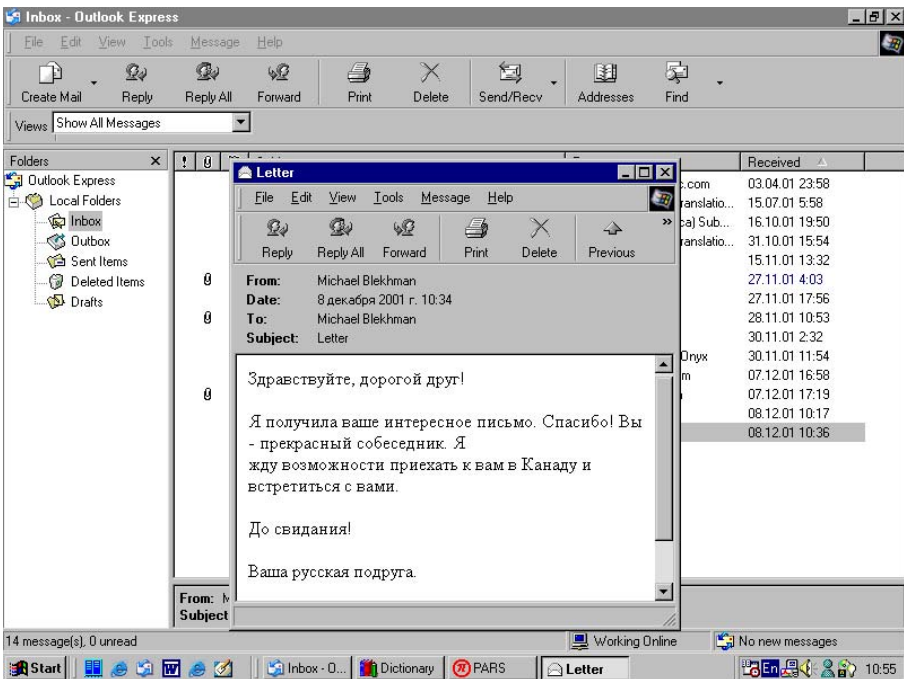
To turn off the translation mode, the customer simply depresses the red circle *Pars* button or *Turn translation off* in *Tools\Pars\Options*.

Translating in the Drag-and-Drop Mode

This mode is especially convenient when the customer has no MS Word installed on their computer, or if he/she doesn't want to run MS Word to translate, for example, an E-mail message. Or the customer has performed translation and wants to insert it into his/her MS Outlook Express to send it as an E-mail message.

To translate in the drag-and-drop mode, one should do the following:

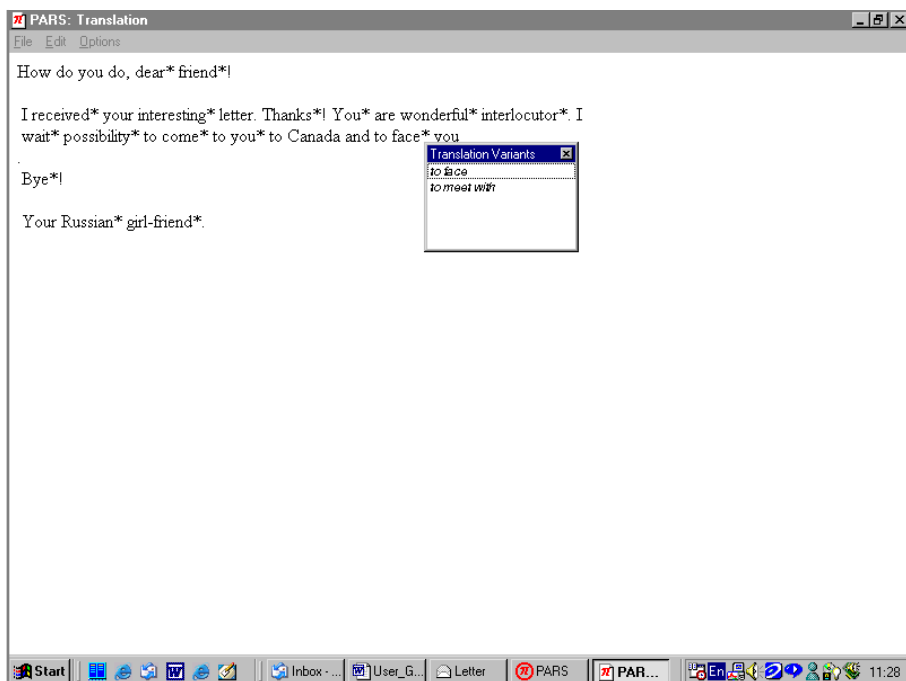
- Go to *Start\Programs\Pars* and click *PARS*. As a result, the so-called 'PARS Shell' will appear on the screen – a panel with PARS commands and options:
- Set up the translation direction and dictionary names in the PARS shell.
- Run any text editor, such as Notepad, WordPad, Word, MS Works, or Outlook Express. The PARS shell will now appear on the bottom panel on the screen.
- Type in or open a text one wants to have translated. For example, the customer has received an E-mail message from a Russian friend:



- To have this message translated into English, the customer will do the following:
 - a. Highlight it.
 - b. Press the *left* mouse button, hold it on the highlighted text and drag it down to the PARS shell.
 - c. Hold it on the PARS shell for a couple of seconds to make the PARS shell “jump” to the top of the screen.

d. Drag the circle up, to the PARS shell at the top of the screen, and drop it there on the *Translate clipboard* button – the left one of the 3 buttons on the PARS shell. (Before dropping, the circle will look as an arrow and a cross).

e. The translation will appear in a separate window: *PARS: Translation*. Again (compare translating in MS Word), each word having multiple translations will be marked with an asterisk. A double click on the asterisk (or, in this mode, on the word itself) will display its translation variants:



To have a text translated *without displaying translation variants*, the customer should have unchecked *Display translation variants* in *PARS Options\General*.

If the customer wants to have his/her English text translated, for example, into Ukrainian, and E-mail it to his/her Ukrainian-speaking correspondent, he/she highlights the English text, drops it as described above on the *Translate clipboard* button in the PARS shell (when the translation direction is English-Ukrainian), and then copies/pastes the translation into Outlook Express.

MSR-MT: The Microsoft Research Machine Translation System

William B. Dolan, Jessie Pinkham, and Stephen D. Richardson

Natural Language Processing Group
Microsoft Research
One Microsoft Way
Redmond, Washington 98052
USA

{billldol, jessiep, steveri}@microsoft.com

Abstract. MSR-MT is an advanced research MT prototype that combines rule-based and statistical techniques with example-based transfer. This hybrid, large-scale system is capable of learning all its knowledge of lexical and phrasal translations directly from data. MSR-MT has undergone rigorous evaluation showing that, trained on a corpus of technical data similar to the test corpus, its output surpasses the quality of best-of-breed commercial MT systems.

1 System Description

MSR-MT is a data-driven MT system that combines rule-based and statistical techniques with example based transfer [1]. MSR-MT has undergone rigorous evaluation showing that, trained on a corpus of technical data similar to the test corpus, its output surpasses the quality of best-of-breed commercial MT systems. In addition, a large pilot study has shown that users judge the system's output accurate enough to be useful in helping them perform highly technical tasks.

Figure 1 below provides a simplified schematic of the architecture of MSR-MT. The central feature of the system's training mode is an automatic logical form (LF) alignment procedure which creates the system's translation example base from sentence-aligned bilingual corpora [2]. During training, statistical word association techniques [3] supply translation pair candidates for alignment and identify certain multi-word terms. This information is used in conjunction with information about the sentences' LFs, provided by robust, broad-coverage syntactic parsers, to identify phrasal transfer patterns.

At run-time, these same syntactic parsers are used to produce an LF for the input string. The goal of the transfer component is thus to identify translations for pieces of this input LF, and to stitch these matched pieces into a target language LF which can serve as input to generation. The example-based transfer component is augmented by decision trees that make probabilistic decisions about the relative plausibility of competing transfer mappings in a given target context [4].

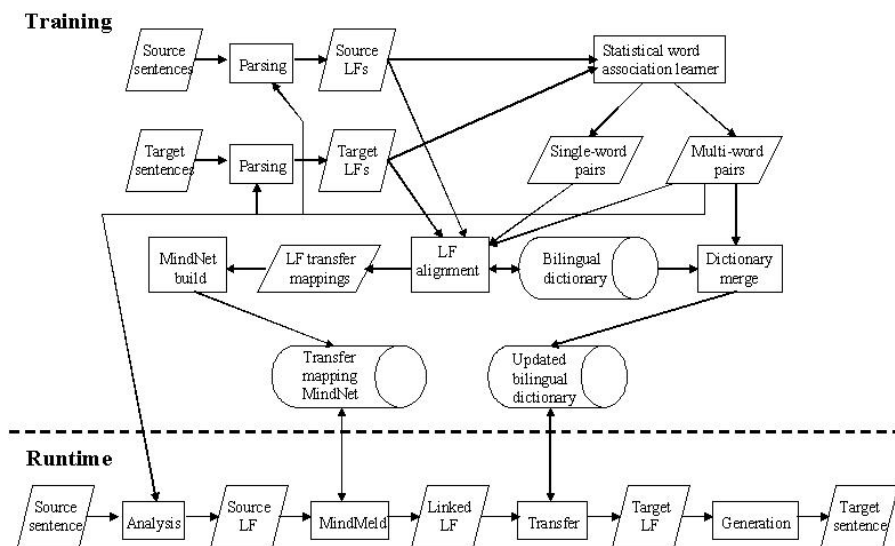


Fig. 1. MSR-MT Architecture (from [1])

The broad-coverage parsers used by MSR-MT were created originally for monolingual applications and have been used in Microsoft Word's grammar checker [5]. Parsers now exist for seven languages (English, French, German, Spanish, Chinese, Japanese, and Korean), and active development continues to improve their accuracy and coverage. These analysis components rely on hand-crafted monolingual lexicons, but the only bilingual lexical resource available to the system is a small lexicon of function word translations; all other lexical translation information is learned automatically from data.

Generation components are currently being developed for English, Spanish, Japanese, German, and French. The English, Spanish, and Japanese generation components have been built by hand, while for German the mapping from logical form to sentence string has been approached as a machine learning problem [6]. A machine-learned generation component for French is also under development.

We have thus far created systems that translate into English from French, German, Spanish, Japanese, Chinese, and Japanese and that translate from English to Spanish, German, and Japanese. MSR-MT's modular architecture means that, in principle, it is possible to rapidly create an MT system for any language pair for which the necessary parsing and generation components exist, along with a suitable corpus of bilingual data. A successful initial experiment in rapidly deploying a new system (French-Spanish) is described in [7]. We have also experimented preliminarily with Chinese to Japanese.

While performance has not been a focus of our research, both training and translation time are fast enough to allow for interactive error analysis and development. Training a transfer database from a 300K sentence bilingual corpus (average sentence length is 18.8 words) takes about 1.5 hours on a cluster of 40 processors averaging

500MHz each and running Windows XP. Run-time translation on a dual-processor 1 GHz PC averages about 0.31 seconds per sentence, or about 59 words per second.

When trained on thousands of bilingual sentence pairs taken from the Microsoft technical domain, MSR-MT has been shown to yield translations that are superior to those produced by best-of-breed commercial MT systems. These claims are based on rigorous evaluations carried out by multiple (typically 6-7) human raters, each examining hundreds of test sentences. These test sentences are drawn from the same pool as the training sentences, and are thus similar in length and complexity. Test and training data are kept strictly separate, and the test data is blind to system developers. Human raters have no knowledge of the internal workings of either MSR-MT or the other MT systems, are employed by an independent vendor organization, and are given no indication of which system produced the translations they are rating. We believe this evaluation methodology to be one of the most rigorous described in the MT literature.

In addition to these ongoing quality evaluations, a pilot study involving 60,000 Spanish users of Microsoft's Spanish language technical support web site found that they were overwhelmingly satisfied with the quality of MSR-MT's translations of English technical documentation. A random sample of approximately 400 users found that 84% were satisfied with the translation quality of the technical articles they accessed.

References

1. Richardson, S., Dolan, W., Menezes, A., Pinkham, J.: Achieving commercial-quality translation with example-based methods. In: *Proceedings of MT Summit VIII* (2001) 293-298
2. Menezes, A., Richardson, S.: A Best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In: *Proceedings of the Workshop on Data-Driven Machine Translation, ACL 2001* (2001) 39-46
3. Moore, R.: Towards a Simple and Accurate Statistical Approach to Learning Translation Relationships Among Words. In: *Proceedings of the Workshop on Data-Driven Machine Translation, ACL 2001* (2001) 79-86
4. Menezes, A.: Better contextual translation using machine learning. In: *Proceedings of the AMTA 2002 Conference* (2002)
5. Heidorn, G. E.: Intelligent Writing Assistance. In: Dale, R., Moisl, H., Somers, H. (eds.): *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*, Marcel Dekker, New York (2000) 181-207
6. Corston-Oliver, S., Gamon, M., Ringger, E., Moore, R.: An overview of Amalgam: a machine-learned generation module. In: *Proceedings of second International Natural Language Generation Conference (INLG)*, Harriman, New York (2002)
7. Pinkham, J., Corston-Oliver, M., Smets, M., Pettenaro, M.: Rapid Assembly of a Large-scale French-English MT system. In: *Proceedings of the MT Summit VIII* (2001) 277-281

The NESPOLE! Speech-to-Speech Translation System

Alon Lavie¹, Lori Levin¹, Robert Frederking¹, and Fabio Pianesi²

¹ Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
`{alavie|ls1|ref}@cs.cmu.edu`

² ITC-irst, Trento, Italy
`pianesi@itc.it`

Abstract. NESPOLE! is a speech-to-speech machine translation research system designed to provide fully functional speech-to-speech capabilities within real-world settings of common users involved in e-commerce applications. The project is funded jointly by the European Commission and the US NSF. The NESPOLE! system uses a client-server architecture to allow a common user, who is browsing web-pages on the internet, to connect seamlessly in real-time to an agent of the service provider, using a video-conferencing channel and with speech-to-speech translation services mediating the conversation. Shared web pages and annotated images supported via a Whiteboard application are available to enhance the communication.

- **Name of System Builders:** research groups of authors above.
- **System Category:** research system.
- **System Characteristics:** Languages: English/German/French to/from Italian. Domain: Travel and Tourism. Networking Features: internet distributed architecture.

1 Introduction

NESPOLE!¹ is a speech-to-speech machine translation system designed to provide fully functional speech-to-speech capabilities within real-world settings of common users involved in e-commerce applications. The project is a collaboration between three European research laboratories (IRST in Trento, Italy; ISL at Universität Karlsruhe (TH) in Germany; and CLIPS at Université Joseph Fourier in Grenoble, France), one US research group (ISL at Carnegie Mellon University in Pittsburgh, PA) and two industrial partners (APT; Trento, Italy – the Trentino provincial tourism board, and Aethra; Ancona, Italy – a telecommunications company). The project is funded jointly by the European Commission and the US NSF.

The main goal of NESPOLE! is to advance the state-of-the-art of speech-to-speech translation in realistic scenarios and involving naive users. The first showcase presented in this demonstration involves an English-, French-, or German-speaking client enquiring about winter-vacation options in the Trentino region

¹ NESPOLE! – NEgotiation through SPOken Language in E-commerce.

of the Italian Alps via a NetMeeting[®] connection². His or her questions are answered by an Italian-speaking agent at APT, while the NESPOLE! system provides speech-to-speech translation and a multi-modal Whiteboard, which users can use to point to shared web-sites or draw on shared maps, therefore enhancing the verbal interaction with further communication capabilities.

The Interlingua-based translation system covers the activities of planning and scheduling winter holidays and similar activities in the Trentino region. By using NESPOLE!, customers can be served in several languages without the need to employ agents capable of speaking the various languages.

2 System Description

2.1 Architecture and Hardware Requirements

The system requires no special hardware on the client's side, except for a standard PC or portable device with microphone and loudspeakers or headsets as well as an Internet connection with a bandwidth of about 64kbit/s. We have, for example, demonstrated the system on a laptop running Windows 2000, connected to the Internet via a wireless LAN link.

The client connects to a special server – the “Mediator” – which in turn establishes connections to the “HLT-servers”, which provide the speech translation capabilities. The Mediator also runs under Windows, while the HLT servers run on different flavors of Linux on Intel PCs or Unix workstations. The IP connections between the Mediator computer and the agent and client use the H323 video-conferencing standard, which is based on UDP for the audio stream. Data is transmitted via TCP. This means that there will be little time delay during transmission, which is important for human-to-human communication, but short segments of speech can be lost during transmission. The links between the Mediator and the HLT servers use TCP. This distributed system design is shown in Figure 1.

The system complexity is hidden from the user, as he only communicates with the Mediator computer. Currently, we usually run the Mediator at IRST in Trento, while the agent is being called at APT, also in Trento. The HLT servers, which provide speech recognition, translation and synthesis, run at the locations of the participating partners, i.e. at the Universities in Trento, Pittsburgh, Grenoble and Karlsruhe. The system design allows for maximum flexibility during usage: Mediators and HLT-servers can be run in several locations, so that the optimal configuration given the current user location and network traffic can be chosen at run-time. The computationally intensive part of speech recognition and translation is done on dedicated server machines. The client machine can therefore be very “thin”, so that the service is available nearly everywhere and to everyone, including mobile devices and public information kiosks.

² The transmission of video is optional and can be suppressed in order to reduce bandwidth, as the system functionality is fulfilled via the data and audio streams.

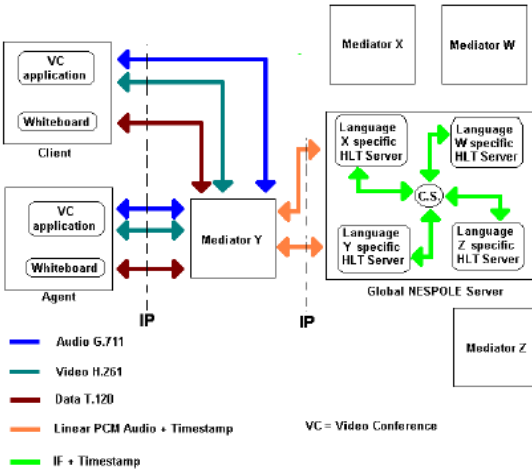


Fig. 1. The Architecture of the NESPOLE! System.

2.2 Software

The speech recognition and translation components have been developed by the participating partners during the project. Further information on the implementation of these modules can be found in the references at the end of this paper. A complete call through the system starts by a client "request" for a NetMeeting connection with the Mediator. The Mediator identifies the client's native language by the language of the originating web page. The Mediator then establishes a connection with the appropriate HLT servers (client's language and Italian, for the agent) and then establishes a NetMeeting connection to the agent. It only accepts the call from the client, once these three required connections have been established. Speech from the client is received by the Mediator, forwarded to the respective HLT server, which in turn performs speech recognition and analysis into a language-independent "Interchange Format" (IF) [5]. The IF is then transmitted to the HLT server associated with the agent, where text is generated. This text string is then synthesized and the resulting audio is transmitted to the agent via the Mediator. Multi-modal gestures, such as drawing on a map or video data are transmitted directly between the communication partners.

System response time is highly variable due to the uncertain and varying network conditions. The speech recognition components use run-on recognition, i.e. recognition starts as soon as the first packets of data arrive, and run approximately real-time (German) or less than 3 times real-time (English) on standard 1GHz Pentium-III PCs running Linux. Depending on network conditions, text representations of speech recognition or translation are available in less than one second after a subject stopped speaking. Under bad network conditions, the same process can, however, take several seconds.

3 Further Information

The NESPOLE! project [4] has already lead to a number of publications on speech recognition [6], [7],[1] and Interlingua-based speech-to-speech translation [5],[3]. The NESPOLE! database is described in [2]; The project web-site can be found at <http://nespole.itc.it>.

4 Acknowledgments

We wish to thank the following members of the NESPOLE! project for substantial contributions to the system described in this paper: Donna Gates, Chad Langley, Kornel Laskowski, Kay Peterson, Tanja Schultz, Alex Waibel and Dorcas Wallace (Carnegie Mellon University); Roldano Cattoni, Gianni Lazzari, Nadia Mana and Emanuele Pianta (ITC-irst); Florian Metze, John McDonough and Hagen Soltau (Universität Karlsruhe); Laurent Besacier, Hervé Blanchon and Dominique Vaufrezydaz (CLIPS, Université Joseph Fourier); Loredana Taddei and Franco Balducci (AETHRA); Erica Costantini (University of Trieste).

The research work reported here was supported by the National Science Foundation under Grant number 9982227 and the European Union under Grant number IST 1999-11562 as part of the joint EU/ NSF MLIAM research initiative.

References

1. L. Besacier, H. Blanchon, Y. Fouquet, J. Guilbaud, S. Helme, S. Mazenot, D. Moraru, and D. Vaufrezydaz: Speech Translation for French in the NESPOLE! European Project. In *Proceedings of EuroSpeech 2001*, Aalborg, Denmark, (2001). 243
2. S. Burger, L. Besacier, P. Coletti, F. Metze, and C. Morel: The NESPOLE! VoIP Dialogue Database. In *Proceedings of EuroSpeech 2001*, Aalborg, Denmark, (2001). 243
3. R. Cattoni, M. Federico, and A. Lavie: Robust Analysis of Spoken Input combining Statistical and Knowledge-based Information Sources. In *Proceedings of the 2001 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2001)*, Madonna di Campiglio, Italy, (2001). 243
4. A. Lavie, F. Balducci, P. Coletti, C. Langley, G. Lazzari, F. Pianesi, L. Taddei, and A. Waibel: Architecture and Design Considerations in NESPOLE!: a Speech Translation System for E-Commerce Applications. In *Proceedings of the 2001 Human Language Technology Conference (HLT-2001)*, San Diego, CA, (2001) 31–34. 243
5. L. Levin, D. Gates, F. Pianesi, D. Wallace, T. Watanabe, and M. Woszczyna: Evaluation of a Practical Interlingua for Task-Oriented Dialogues. In *Proceedings of Workshop On Applied Interlinguas: Practical Applications of Interlingual Approaches to NLP, at ANLP/NAACL-2000 Conference*, Seattle, WA, (2000) 18–23. 242, 243
6. F. Metze, J. McDonough, and H. Soltau: Speech Recognition over NetMeeting Connections. In *Proceedings of EuroSpeech 2001*, Aalborg, Denmark, (2001). 243
7. D. Vaufrezydaz, L. Besacier, C. Bergamini, and R. Lamy: From generic to task-oriented speech recognition: French experience in the NESPOLE! European project. In *Proceedings of ITRW Workshop on Adaptation Methods for Speech Recognition*, Sophia Antipolis, France, (2001). 243

The KANTOO MT System: Controlled Language Checker and Lexical Maintenance Tool

Teruko Mitamura, Eric Nyberg, Kathy Baker, Peter Cramer
Jeongwoo Ko, David Svoboda, and Michael Duggan

Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{teruko, ehnl}@cs.cmu.edu

1 Introduction

We will present the KANTOO machine translation environment, a set of software servers and tools for multilingual document production. KANTOO includes modules for source language analysis, target language generation, source terminology management, target terminology management, and knowledge source development (see Figure 1).

KANTOO is a knowledge-based, interlingual machine translation system for multilingual document production. KANTOO includes: a) an MT engine, the result of fundamental redesign and reimplementations of the core algorithms of the KANT system [2,4]; and b) a set of off-line tools that support the creation and update of terminology and other knowledge resources for different MT applications. The demonstration will focus on two of the newer capabilities in KANTOO:

- **Controlled Language Checker (CLC).** The CLC is a thin Java client which supports interactive editing and automatic checking of XML documents. This tool performs vocabulary and grammar checking on each sentence in a document. The checker accesses the KANTOO Analyzer (running as a separate network service), which performs tokenization, morphological processing, lexical lookup, syntactic parsing and semantic interpretation. Wherever possible, the system tries to resolve problems (e.g., ambiguity) automatically without troubling the user. If a sentence does not pass the check, then a diagnostic message is produced for the user to resolve. For example, the user might be asked to choose between two candidate meanings for a term. If the sentence does not conform to the controlled language, the system may present an acceptable rewrite which the user can select with one click.
- **Lexical Maintenance Tool (LMT)** The LMT was originally implemented as an Oracle database and Forms application, and has been recently redesigned using a Java web interface to Oracle. The Java web interface has a distinct advantage is a multi-user PC environment where it is not usual for Oracle Forms to be installed. The LMT allows the terminology maintainer

to create, modify and navigate through large numbers of lexical entries. The LMT brings together the various kinds of lexical entries used in MT development, such as words, phrases and specialized entries like acronyms, abbreviations, and units of measure. The system also includes a batch load process which allows rapid, automatic definition of new technical terminology from filled templates.

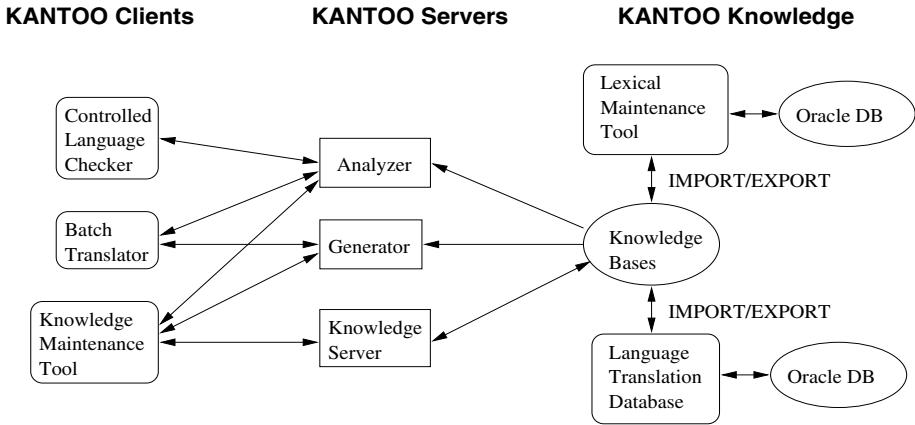


Fig. 1. KANTOO Architecture.

The KANTOO architecture is scalable; several domains, languages, and versions of their knowledge sources can be maintained and executed in parallel. The tools are in daily use at an industrial document production facility [1]. Recent improvements to the system, which involve creation of a client/server architecture for Controlled Language Checking and terminology maintenance, are intended to expand the utility of the system to environments where work is outsourced to vendors with access to the corporate network via Windows PCs.

2 Other KANTOO Modules

- **Analyzer.** The Analyzer module performs tokenization, morphological processing, lexical lookup, syntactic parsing with a unification grammar, and semantic interpretation, yielding one or more interlingua expressions for each valid input sentence (or a diagnostic message for invalid sentences). The same

Analyzer server can be used simultaneously by the CLC, Batch Translator and KMT¹.

- **Generator.** The Generator module performs lexical selection, structural mapping, syntactic generation, and morphological realization for a particular target language. The same Generator executable can be loaded with different knowledge bases for different languages. The same Generator server can be used by the Batch Translator and KMT in parallel.
- **Language Translation Database (LTD).** The LTD is the target language counterpart to the LMT, and is implemented using Oracle and Forms. The LTD includes productivity enhancements which provide the translator with partial draft translations taken from similar translated terms.
- **Knowledge Maintenance Tool (KMT) and Knowledge Server.** The Knowledge Maintenance Tool (KMT) is a graphical user interface which allows developers to test their knowledge changes in the context of a complete working system. Users can trace or edit individual rules or categories of rules. The KMT operates in conjunction with the Knowledge Server, which provides distributed network access to a version-controlled repository of KANTOO knowledge sources.

3 Contents of the Proposed Demonstration

We plan to demonstrate the KANTOO system in three modes:

1. **Controlled Language Checking.** We will demonstrate the Controlled Language Checker on some sample technical texts which illustrate the various kinds of feedback and diagnostic messages that the CLC can provide. The user can open an XML document using the tool, which will either check all sentence in the text on demand (batch mode), or check each sentence that is edited or typed in (interactive mode). The system uses color and hyperlinks to call attention to parts of the text which require intervention by the user (ambiguities, unknown terminology, ungrammatical sentences, etc.). The system can also provide term lookup information to help the user resolve ambiguous usage.
2. **Automatic Translation.** The texts that are checked will be translated to Spanish as part of a live demonstration. The checking tool includes a Translate menu command, which pops up a new editor window containing the translated text, which can be saved to a separate file. The checking tool interacts with the networked Analyzer and Generator servers in order to carry out the translation.
3. **Terminology Update.** We will demonstrate how the LMT interface can be used to browse and update technical terminology used by the system. The LMT supports wildcard search for technical terms in various lexical

¹ Space limitations preclude a discussion of a) the Controlled Language Checker, which has been discussed at length in [3], and b) the Batch Translator, which is a simple piece of driver code that uses the KANTOO servers to translate entire documents.

categories, and provides complete access to all term features via a set of editing panels in the user interface.

4 Additional Information

- **System Builder / Contact:** Teruko Mitamura and Eric Nyberg, Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213, {teruko,ehn}@cs.cmu.edu.
- **System Category:** Research and development system, with commercial licensing and deployment for industrial partners. Several KANTOO modules (including the Lexical Maintenance Tool, Controlled English, French, Spanish and German) have been deployed at Caterpillar, Inc. [1].
- **System Characteristics:** The Controlled Language Checker (CLC) is a Java client program which interacts with the KANTOO Analyzer (a networked Unix server). The Lexical Maintenance Tool (LMT) includes a networked Oracle database and a separate Java user interface. KANTOO also supports domain-specific MT for Controlled English to Spanish, French, and German. Core systems for Portuguese, Italian and Polish have been developed but not deployed. Controlled language checker operates as a client program (Java). The Analyzer and Generator components in the MT system operate as networked servers (Unix).

References

1. Kamprath, C., Adolphson, E., Mitamura, T. and Nyberg, E.: Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English. In: Proceedings of the Second International Workshop on Controlled Language Applications (1998) 245, 247
2. Mitamura, T., Nyberg, E. and Carbonell, J.: An Efficient Interlingua Translation System for Multi-lingual Document Production. In: Proceedings of the Third Machine Translation Summit (1991) 244
3. Mitamura, T. and Nyberg, E.: Controlled English for Knowledge-Based MT: Experience with the KANT System. In: Proceedings of TMI-95 (1995) 246
4. Nyberg, E. and Mitamura, T.: The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains. In: Proceedings of COLING-92 (1992) 244

Many other publications about the KANTOO system can be found on the KANT Home Page: <http://www.lti.cs.cmu.edu/Research/Kant>

Approaches to Spoken Translation¹

Christine A. Montgomery
Naicong Li

Language Systems Inc.
Woodland Hills, California
{chris, naicong}@lsi.com

Abstract. The paper discusses a number of important issues in speech-to-speech translation, including the key issue of level of integration of all components of such systems, based on our experience in the field since 1990. Section 1 discusses dimensions of the spoken translation problem, while current and near term approaches to spoken translation are treated in Sections 2 and 3. Section 2 describes our current expectation-based, speaker-independent, two-way translation systems, and Section 3 presents the advanced translation engine under development for handling spontaneous dialogs.

1 Dimensions of the Problem

The complexity of integrating all the required components for speech-to-speech translation is dependent upon the desired level of integration that is to be achieved among components. Ideally, the components of such a system should be capable of functioning as a human does, trying different strategies in parallel to make sense of a stream of speech such as "Note robbery was taken from these rooms" to derive "No property was taken from these rooms" and to generate a correct translation in another language. Humans apply simultaneously all of their phonetic, morphological, syntactic, semantic, and pragmatic knowledge to analyze input utterances and generate corresponding output utterances in other languages. They do this by interleaving knowledge from the different components to generate a hypothesis about the content of an utterance based on the knowledge represented in one component—say, phonology—to produce "Note robbery", subsequently rejecting that hypothesis as untenable based on the knowledge of the semantic component that "robbery" is an abstract object that cannot be removed from a location. Then another strategy is tried, generating a hypothesis of a construct of a similar phonetic shape that can be removed from a location, is within the proper domain, and is a plausible utterance in the discourse context. Thus "No property" can be derived, calling on all these different types of knowledge, more or less in parallel.

Some attempts have been made to emulate these aspects of human speech processing behavior, the earliest being the HearsayII system [1], and more recently, the

¹ The work reported in this paper was partially supported by the National Institute of Standards and Technology under an Advanced Technology Project Cooperative Agreement (No. 70NANB8H4055).

Verbmobil project [2]. However, achievement of this level of integration of all components for a speech-to-speech translation system is still the exception rather than the rule, because of the enormous complexity of the task.

Just the construction of a linguistically based spoken translation system incorporating all these components with a minimal level of integration is a challenging task, as discussed in LSI's paper in the 1994 AMTA proceedings [3] describing a research prototype which provided two-way translation for Spanish, Arabic, and Russian in a limited domain. This system analyzed spoken source input into an interlingual representation and generated spoken target translations from the interlingua, which was based on Jackendoff's lexical conceptual structures [4] and LSI's discourse structures.

Evaluation of systems of this complexity, including all the NLP components and the speech components, is equally daunting, as shown in our analysis of an earlier, somewhat less complex, spoken translation system [5]. In a two-way spoken translation system, there are two recognizers running simultaneously, adding the problem of wrong language selection to the inventory of possible recognition errors [6, 7]. Our more recent work has added another level of complexity in providing transcription of the content of two-way spoken interviews into electronic forms [8].

To cope with erroneous inputs in an earlier NLP system for processing military messages, we utilized "Unexpected Inputs" modules in the character processing, morphological, syntactic, and semantic components [9]. However, ideally, handling the type of conversational inputs ultimately envisioned in spoken translation systems research will require much more, including rejection of erroneous word hypotheses generated by the ASR systems, as well as dealing with indications of emotion, hesitations, continuations, instances of self-repair, ungrammatical utterances, and the like, generated by the participants in the conversation.

These are some rough dimensions of the problems involved in speech-to-speech processing from our own experience with such systems. However, rather than dwell on the complexity of the ultimate solutions to all of the speech and language processing problems that are associated with spoken translation systems, this paper presents some nearer term and partial solutions to these problems, which will allow continued progress toward the ideal.

2 Spoken Translation Using Expectation-Based Dialogs

The spoken translation systems we have developed for medical and law enforcement applications are comprised of the following components: *the speech recognition component, the language understanding and translation component, the speech generation component*, and for our speech-to-forms software, *the forms component*.

The systems are speaker-independent, two-way translation systems for conducting interviews between English-speaking law enforcement or medical personnel and the non-English speaking public. The most substantial development work has been carried out on English-Spanish, but there has also been a considerable amount of development for Mandarin Chinese, Korean, and Russian.

The systems have an open microphone (they are not push-to-talk or prompt-driven), two recognizers are always active at the same time, and the dialogs are mixed initiative--either speaker can begin speaking at any time. Thus, if a police officer says "You have the right to consult an attorney", the Hispanic respondent, after hearing the Spanish translation of the English input, may say "*No entiendo*" - "I don't understand", or ask for the phrase to be repeated, "*Otra vez, por favor*", or interject other utterances. The dialogs are partitioned by discourse topic and subtopic in order to improve accuracy: for collecting biographic data, there are separate dialogs for name, address, zip code, and telephone number, and other such information.

The dialogs are expectation-based; the set of allowable English commands and questions and possible Spanish responses are derived from observations of actual interviews, recorded interviews, or simulated interviews with volunteer respondents and professional interviewers (e.g., a hospital admissions clerk). Each dialog topic is comprised of a dialog pair, consisting of the set of possible English queries and commands for that particular topic or subtopic in the restricted domain of the given application, and the set of possible Spanish responses. In order to improve the habitability of these dialogs for both participants, a number of different ways of requesting information and answering questions is provided in the dialogs. For example, an officer may ask any of the following questions to obtain information on the respondent's nationality:

What is your nationality?

What country were you born in?

Which country were you born in?

In what country were you born?

In which country were you born?

Where were you born?

Where are you from?

What country are you from?

There are two types of questions here, one which requires a country name represented by a proper noun as an argument in the response, and one which expects a nationality argument represented by predicate adjective. Thus the response set for the dialog must include answers of both types:

Soy de Mexico.

Mexico.

Nací en Chile.

Soy guatemalteca.

Guatemalteca.

Some of our current research work is aimed at improving the accuracy of our expectation-based speech-to-speech and speech-to-forms translation software for law enforcement and medical applications, using dialog structures such as these. We are currently working on a number of improvements to the dialog management component. These will take advantage of the expectation-based dialog structure to improve accuracy for both English and foreign language recognizers. They are based on utilization of conversational analysis (CA) principles, e.g., turn-taking, adjacency pairs [10], which are now only partially exploited. All of the dialogs have more than one possible path that may be taken, depending on which questions are used, and in which sequence, and whether the interviewer and/or the respondent inserts utterances from the background dialog, e.g., "*No entiendo*" or "Please speak louder". We are devel-

oping a Dialog Tracker, which will trace the path the dialog is taking, and determine whose turn it is (i.e., which speaker is most likely to speak next), and thus, which recognizer should be called first at that point in the dialog. A second set of improvements will exploit the notion of adjacency pairs [10]. Specific adjacency pair information can also be used for contextual disambiguation for items such as *¿hace?* in questions and responses like the following, where a time period must be distinguished from a time point:

How long have you been on insulin?

For five years.

When was the last time you took your insulin?

Five hours ago.

¿Hace cuánto tiempo que toma insulina?

Hace cinco años.

¿Cuándo fué la última vez que tomó su insulín

Hace cinco horas.

3 An Approach for Handling Spontaneous Dialogs

Much of our current research work is aimed at implementing a more advanced translation engine than the expectation-based engine described above to be used for two-way translation of spontaneous narrative and dialogs. This engine, LSITrans, is based on the research prototype described in [1], which analyzed source language utterances into an interlingua and generated target language utterances from it [3, 11]. In the current development, we have moved toward a transfer-based rather than an interlingua-based approach, in order to streamline the processing, and reduce the required amount of linguist hand-crafting involved in construction of the linguistic and domain knowledge bases. Language pairs currently under development are English-Spanish, English-Korean, and English-Russian.

As in the earlier system, each step in the translation process involves a particular module with its own set of rules which creates a data structure that, in turn, feeds the following module until the initial source sentence is mapped into a target sentence. Additionally, there is a module of lexical resources (composed of a source language lexicon, a concept knowledge-base, and a target language lexicon) that is accessed at different points in the translation process. Following the lexical lookup, the source utterance is analyzed by a chart parser, which produces one or more syntactic analyses. These analyses serve as input for the next module, the Functional Parse (FP), which analyzes the syntactic tree in terms of syntactic type and semantic functions of the elements.

To do this, the FP module reads a set of language dependent rules that map from the syntactic structure to the specific slots of this representation.² The FP extracts and sets up information concerning clause type, main predicate, argument structure in terms of grammatical roles, relations within the noun phrase and the ontologi-

² FP rules take the form of tree navigation and information retrieval instructions. Given a specific type of sentence, the FP rules determine which nodes of the syntactic representation are searched for and which information is transferred to the FP.

cal type of each of its objects. This representation combines syntactic and semantic information. The next step maps the source FP into a target FP. The compositional translation engine then applies a number of transfer rules to the source FP yielding an FP representation that is the basis for the translation. In cases where translation divergences exist, it is necessary to apply transfer rules, which are mainly language pair dependent (e.g., English-to-Spanish), although some transfer rules (e.g., copula deletion) may apply to more than one language pair. The changes a transfer rule specifies may involve change of concepts, change of verb class (this includes different types of divergences related to grammatical and semantic roles), and addition or deletion of FP nodes, among other things.

The last module of the translation engine generates the surface form of the target sentence. The generation process involves four main steps: lexical selection, application of target language word ordering rules, replacement of base forms with proper inflected forms, and propagation of agreement features. A detailed description of this translation engine is given in [8].

References

1. Erman, L. D., Hayes-Roth, F., Lesser, V. R and D. R. Reddy: The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys* 12(2) (1980) 213–253
2. Wahlster, W. (Ed.): *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer-Verlag (2000)
3. Stalls, B. G., Belvin, R. S., Arnaiz, A. R., Montgomery, C. A., and Stumberger, R. E.: An adaptation of lexical conceptual structure to multilingual processing in an existing text understanding system. In: *Proceedings, AMTA (1994)* 106-113
4. Jackendoff, R.: *Semantics and Cognition*. MIT Press (1983)
5. Montgomery, C. A., Stalls, B. G., Stumberger, R. E., Li, N., Belvin, R. S., and Arnaiz, A. R.: Evaluation of the Machine-Aided Voice Translation (MAVT) System. In: *Proceedings of the Workshop on Machine Translation Evaluation (1994)* 27-30
6. Montgomery, C. A. and Crawford, D. J.: ELSIE, The Quick Reaction Spoken Language Translator (QRSLT). In: *The Information Age, Proceedings of the Dual-Use Technologies and Applications Conference, IEEE (1997)* 39-43
7. Montgomery, C. A. and Crawford, D. J.: ELSIE, The Quick Reaction Spoken Language Translator (QRSLT), Final Technical Report (AFRL-IF-RS-2000-106) (2000)
8. Montgomery, C. A., Li, N., Lee, M., Zarazúa, D., Kass, M., and M. Chung: Spoken Language Forms Translation for Medical Interviews. Final Technical Report, NIST ATP Project (in preparation, 2002)
9. Montgomery, C. A. and Stalls, B. G.: A Sublanguage for Reporting and Analysis of Space Events. In: Grishman and Kittredge (eds.): *Analyzing Language in Restricted Domains: Sublanguage description and processing*. Lawrence Erlbaum, NJ (1986)
10. Sacks H., Schegloff, E. and Jefferson, G.: A Simplest Systematics for the Organization of Turn-taking in Conversation. In: *Language* 50 (1974) 696-735
11. Montgomery, C. A., Stalls, B. G., Stumberger, R. E., Li, N., Belvin, R. S., Arnaiz, A. R. and Litenatsky, S. H.: The Machine-Aided Voice Translation System. In: *Proceedings, AVIOS (1995)* 109-111

Author Index

Abir, Eli,	216	Langlais, Philippe	44, 104
Akers, Glenn A.	220	Lapalme, Guy	44
Baker, Kathryn	145, 244	Lavie, Alon	1, 240
Bender, Howard J.	224	Lee, Hyo-Kyung	114
Benjamin, Bryce	229	Levin, Lori	1, 240
Blekhman, Michael	232	Li, Naicong	248
Brown, Ralf	1	Maier, Elisabeth	187
Carbonell, Jaime	1	Manandise, Esmé	64
Carl, Michael	11	Marcu, Daniel	155, 229
Casacuberta, Francisco	54	Menezes, Arul	124
Chang, Jason S.	21	Miller, David	216
Chiba, Yasunobu	165	Mitamura, Teruko	145, 244
Choi, Sung-Kwon	94	Monson, Christina	1
Chuang, Thomas C.	21	Montgomery, Christine A.	248
Clarke, Anthony	187	Moore, Robert C.	135
Cramer, Peter	244	Morland, Verne	195
Dolan, William B.	237	Ney, Hermann	54
Dorr, Bonnie J.	31, 84	Nyberg, Eric	145, 244
Duggan, Michael	244	Och, Franz J.	54
Foster, George	44	Pearl, Lisa	31
Frederking, Robert	240	Peterson, Brian	145
García-Varea, Ismael	54	Peterson, Erik	1
Gdaniec, Claudia	64	Pianesi, Fabio	240
Gough, Nano	74	Pinkham, Jessie	237
Habash, Nizar	31, 84	Probst, Katharina	1
Hamamoto, Takeshi	165	Rakova, Alla	232
Hearne, Mary	74	Richardson, Stephen D.	237
Hong, Munpyo	94	Rychtyckyj, Nestor	207
Horiuchi, Takashi	165	Schäler, Reinhard	11
Huang, Yinxia	94	Seo, Young Ae	94
Hwa, Rebecca	31	Simard, Michel	104
Kim, Changhyun	94	Soricut, Radu	155
Kim, Young Ki	94	Stadler, Hans-Udo	187
Klein, Steve	216	Steinbaum, Michael	216
Knight, Kevin	155, 229	Svoboda, David	145, 244
Ko, Jeongwoo	244	Utsuro, Takehito	165
Kursin, Andrei	232		

Way, Andy	11, 74
Weerasinghe, Ruwan	177
Williams, Jennifer	145
Yang, Sung Il	94
You, GN	21