Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation

TMI 2007

Skövde University Studies in Informatics
2007:1

Skövde University Studies in Informatics

SUSI is a series of publications published by the School of Humanities and Informatics at University of Skövde. It covers all aspects of theoretical and applied informatics (in a broad sense) research in the form of monographs, doctoral dissertations, textbooks, and proceedings volumes.

Series editor: Anders Malmsjö

Volume 2007:1

Recently published in this series

Vol. 2005:1 S.F. Andler, A. Cervin (Eds.), RTIS 2005 - Proceedings of Real Time in Sweden 2005, the 8th Biennial SNART Conference on Real Time Systems.

Vol. 2005:2 P. Backlund, S. Carlsson, E. Söderström (Eds.), BIR 2005 - Proceedings of the 4th International Conference on Business Informatics Research, Skövde, 3 – 4 October 2005.

Vol. 2006:1 S. Carlsson, B. Cronquist, H. Kjellin, B. Wangler (Eds.), Knowledge in Organisations I: Foundations and Methodologies, Development and Design, Applications and Integration; Proceedings of the KiO Network, June 2006.

Vol. 2006:2 S. Carlsson, B. Cronquist, H. Kjellin, B. Wangler (Eds.), Knowledge in Organisations II: Foundations and Methodologies, Development and Design, Applications and Integration; Proceedings of the KiO Network, June 2006.

ISSN 1653-2325

Andy Way, Barbara Gawronska (Eds.)

TMI 2007

Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation

A. Way, B. Gawronska (Eds.), *TMI 2007*

Skövde University Studies in Informatics 2007:1
ISSN 1653-2325
ISBN 978-91-977095-0-7

Distribution:
University of Skövde
School of Humanities and Informatics
P.O. Box 408
SE-541 28 Skövde
SWEDEN
+46 (0)500 448000

www.his.se

Printed at Skövde All-kopia, 2007

TMI-07

11th International Conference on Theoretical and Methodological Issues in Machine Translation

Notes from the Programme Chair

The first International Conference on Theoretical and Methodological Issues in Machine Translation took place in upstate New York in 1985. More than 20 years later, we are pleased to be holding the 11th Conference in the series in Skövde, Sweden, from 7—9 September 2007.

As Programme Chair, I was absolutely thrilled that we received 63 submissions to the conference. From this, following a huge amount of hard work by the Programme Committee (listed on the next page), this has been whittled down to the programme that stands before you, consisting of 17 oral presentations, and 12 posters and demonstrations. As is traditionally the case at TMI, we have papers on a wide range of subjects, including statistical MT, example-based MT, rule-based MT, hybrid MT, MT evaluation, open source MT, alignment, inducing bilingual lexical information, parallel and comparable corpora, as well as multilingual applications.

In addition, we have two renowned keynote speakers in Anna Sågvall-Hein (Uppsala) and Hermann Ney (RWTH Aachen). Finally, we have a panel session chaired by Steven Krauwer (Utrecht) entitled *Is MT in Crisis?* which promises to be both fun and informative.

There are a number of people I would like to thank. Firstly, this conference would not be happening at all without the enormous effort provided by Barbara Gawronska and her team in Skövde. We would especially like to thank the School of Informatics for having the Proceedings printed for us. Secondly, my heartfelt thanks go to the members of the Programme Committee, who, as usual, did sterling work over and above what might have reasonably been expected from them. Thirdly, as you all know we decided a long time ago to co-locate the conference temporally with MT Summit XI, so I would like to thank Bente Maegaard and Viggo Hansen for their support in making this possible, and for supporting the discounted registration rates for both conferences. Finally, I would like to thank my students Karolina Owczarzak, Yanjun Ma, John Tinsley and Sara Morrissey for their help in preparing the website, proceedings and letters of support to enable attendees to travel.

We have an excellent programme assembled. I hope you enjoy what you hear.

Andy Way.

TMI-07

11th International Conference on Theoretical and Methodological Issues in Machine Translation

Programme Chair

Andy Way (DCU, Dublin).

Programme Committee

Ralf Brown (CMU, Pittsburgh)
Chris Callison-Burch (Edinburgh)
Michael Carl (IAI, Saarbrücken)
Colin Cherry (Alberta, Edmonton)
Andreas Eisele (DFKI, Saarbrücken)
David Farwell (UPC, Barcelona)
Marcello Federico (ITC-IRST, Trento)
Mikel Forcada (Alacant)
Bob Frederking (CMU, Pittsburgh)
George Foster (NRC, Quebec)
Declan Groves (Traslán, Dublin)
Nizar Habash (Columbia, New York)
Keith Hall (JHU, Baltimore)
Mary Hearne (DCU, Dublin)
John Hutchins (UEA, Norwich)
Rebecca Hwa (Pittsburgh)
Kevin Knight (ISI, Marina Del Rey)
Philipp Koehn (Edinburgh)
Jonas Kuhn (Potsdam, Berlin)
Philippe Langlais (RALI, Montreal)
Alon Lavie (CMU, Pittsburgh)
Yves Lepage (Caen)
Lluís Màrquez (UPC, Barcelona)
Evgeny Matusov (RWTH, Aachen)
Sara Morrissey (DCU, Dublin)
Miles Osborne (Edinburgh)
Michael Paul (ATR, Kyoto)
Fred Popowich (SFU, Burnaby)
Anna Sågvall Hein (Uppsala)
Khalil Sima'an (Amsterdam)
Harold Somers (Manchester)
Oliver Streiter (Kaohsiung, Taiwan)
Nicolas Stroppa (DCU, Dublin)
Nicola Ueffing (NRC, Quebec)
Stephan Vogel (CMU, Pittsburgh)
Haifeng Wang (Toshiba, Beijing)
Dekai Wu (Hong Kong)
Muyun Yang (Harbin, China)

TMI-07

11th International Conference on Theoretical and Methodological Issues in Machine Translation

Local arrangements committee

Chair: Barbara Gawronska (HiS, Skövde)

Björn Erlendsson (HiS, Skövde)
Niklas Torstensson (HiS, Skövde)

TMI-07

11th International Conference on Theoretical and Methodological Issues in Machine Translation

Programme

Thursday 6th September

10.30 – 18.00 Conference excursion

18.00 – 20.00 Conference Registration

19.00 – 21.00 Welcome Reception

Friday 7th September

8.00 – 9.30 Conference Registration

9.15 – 9.45 Introductory Remarks

9.45 – 10.45 Invited Talk: *Rule-based and Statistical Machine Translation with a Focus on Swedish* -- Anna Sågvall-Hein

10.45 – 11.15 COFFEE

11.15 – 11.50 *How Phrase Sense Disambiguation outperforms Word Sense Disambiguation for Statistical Machine Translation* -- Marine Carpuat and Dekai Wu

11.50 – 12.25 *Exploiting Source Similarity for SMT using Context-Informed Features* -- Nicolas Stroppa, Antal van den Bosch and Andy Way

12.25 – 13.00 *Towards Hybrid Quality-Oriented Machine Translation. On Linguistics and Probabilities in MT* -- Stephan Oepen, Erik Velldal, Jan Tore Lønning, Paul Meurer and Victoria Rosen

13.00 – 14.15 LUNCH

14.15 – 14.50 *BLEUATRE: Flattening Syntactic Dependencies for MT Evaluation* -- Dennis Mehay and Chris Brew

14.50 – 15.25 *A Cluster-Based Representation for Multi-System MT Evaluation* -- Nicolas Stroppa and Karolina Owczarzak

15.25 – 16.00 *Combining translation models in statistical machine translation* -- Jesus Andrés-Ferrer, Ismael García-Varea and Francisco Casacuberta

16.00 – 16.30 COFFEE

16.30 – 17.05 *A Greedy Decoder for Phrase-Based Statistical Machine Translation* -- Philippe Langlais, Alexandre Patry and Fabrizio Gotti

17.05 – 17.40 *Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model* -- Kay Rottmann and Stephan Vogel

Saturday 8th September

9.15 – 9.50 *Learning Bilingual Semantic Frames: Shallow Semantic Parsing vs Semantic Role Projection* -- Pascale Fung, Zhaojun Wu, Yongsheng Yang and Dekai Wu

9.50 – 10.15 *Reducing Human Assessment of Machine Translation Quality to Binary Classifiers* -- Michael Paul, Andrew Finch and Eiichiro Sumita

10.15 – 10.50 *A New Method for the Study of Correlations between MT Evaluation Metrics and Some Surprising Results* -- Paula Estrella, Andrei Popescu-Belis and Maghi King

10.50 – 11.20 COFFEE

11.20 – 11.55 *Alignment-Guided Chunking* -- Yanjun Ma, Nicolas Stroppa and Andy Way

11.55 – 12.30 *Sub-Phrasal Matching and Structural Templates in Example-Based MT* -- Aaron Phillips

12.30 – 13.05 *Extracting Phrasal Alignments from Comparable Corpora by Using Joint Probability SMT Model* -- Tadashi Kumano, Hideki Tanaka and Takenobu Tokunaga

13.05 – 14.20 LUNCH

14.20 – 15.50 Panel Session: *Is MT In Crisis?* (Moderator: Steven Krauwer)

15.50 – 16.20 COFFEE

16.20 – 17.50 Demonstration & Poster Session

- *Demonstration of the German to English METIS-II MT System* -- Michael Carl, Sandrine Garnier and Paul Schmidt
- *Automatic induction of shallow-transfer rules for open-source machine translation* -- Felipe Sánchez-Martínez and Mikel Forcada
- *Support Vector Machine Based Orthographic Disambiguation* -- Eiji Aramaki, Takeshi Imai, Kengo Miyo and Kazuhiko Ohe
- *EBMT Based on Finite State Automata Transfer Generation* -- Ren Feiliang, Zhang Li, Hu Minghan and Yao Tianshun
- *Demonstration of the Spanish to English METIS-II MT system* -- Maite Melero and Toni Badia
- *An Assessment of Language Elicitation without the Supervision of a Linguist* -- Alison Alvarez, Lori Levin, Robert Frederking and Jill Lehman
- *Phrase Alignment Based on Bilingual Parsing* -- Akira Ushioda
- *Demonstration of the Dutch-to-English METIS-II MT System* -- Peter Dirix, Vincent Vandeghinste and Ineke Schuurman
- *Hand in Hand: Automatic Sign Language to Speech Translation* -- Daniel Stein, Philippe Dreuw, Hermann Ney, Sara Morrissey and Andy Way
- *Combining Resources for Open Source Machine Translation* -- Eric Nichols, Francis Bond, Darren Scott Appling and Yuji Matsumoto
- *Demonstration of the Greek to English METIS-II MT System* -- Sokratis Sofianopoulos, Vassiliki Spilioti, Marina Vassiliou, Olga Yannoutsou and Stella Markantonatou

- *Reordering via N-Best Lists for Spanish-Basque Translation* -- Germán Sanchis and Francisco Casacuberta

Sunday 9th September

9.15 – 10.15 Invited Talk: *Statistical MT from TMI-1988 to TMI-2007: What Has Happened?* -- Hermann Ney

10.15 – 10.50 *Theoretical and Methodological Issues Regarding the Use of Language Technologies for Patients with Limited English Proficiency* -- Harold Somers

10.50 – 11.20 COFFEE

11.20 – 12.55 *Capturing Translational Divergences with a Statistical Tree-to-Tree Aligner* -- Mary Hearne, John Tinsley, Andy Way and Ventsislav Zhechev

12.55 – 12.30 *Breaking the barrier of context-freeness. Towards a linguistically adequate probabilistic dependency model of parallel texts* -- Matthias Buch-Kromann

12.30 – 13.05 Final Remarks

13.05 – 14.30 LUNCH

15.00 Bus to Copenhagen for *MT Summit XI* arriving at 20.30 (approx.)

TMI-07

11th International Conference on Theoretical and Methodological Issues in Machine Translation

Invited Talks

Rule-based and Statistical Machine Translation with a Focus on Swedish

Anna Sågvall Hein
Professor of Computational Linguistics,
Department of Linguistics and Philology
Uppsala University
P.O. Box 635
SE-751 26 Uppsala
Sweden

anna@lingfil.uu.se

In the talk, I will discuss the pros and cons of rule-based (RBMT) versus statistical machine translation (SMT). In particular, I will present data from running one system of each kind, Convertus and Pharaoh, on a corpus of automotive service literature from Scania CV AB. Translation goes from Swedish to English. The focus will be on the different kinds of errors that are typical of the two approaches, and how the errors may be identified and corrected. Some of them are language-independent whereas others are typical of the language pair in focus.

In addition, data from running Pharaoh on Europarl involving Swedish in relation to Danish, English, German, French, Spanish, Dutch, Portuguese, Italian, Greek and Finnish will be brought into the discussion, and from running Convertus on a corpus of university syllabi from Uppsala University. In the Europarl case, an RBMT system trained for the domain was not available, and in the syllabus case, a parallel corpus for the domain was not available. This seems to be the typical case in a situation where MT is needed. The choice between an RBMT and an SMT approach is constrained by what is available in terms of corpora for an SMT system, and language resources for an RBMT system. I will conclude by discussing how the two approaches may be combined.

Statistical MT from TMI-1988 to TMI-2007: What Has Happened?

Hermann Ney
Professor of Human Language Technology and Pattern Recognition,
RWTH Aachen University
Ahornstr. 55
DE-52056 Aachen, Germany

ney@informatik.rwth-aachen.de

When Peter Brown of IBM research presented a statistical approach to French—English MT at TMI 1988 at CMU, the audience was shocked because this approach was a slap in the face for the then received MT theories. At the time of TMI 2007, nearly two decades later, the statistical approach seems to be the mainstream approach in MT research.

Since the first approach to statistical MT had been worked out by IBM for French—English translation, many attempts have been made to push the state of the art and to improve the translation accuracy.

Statistical MT systems are now able to translate across a wide variety of language pairs and translation tasks. The statistical approach forms the basis for many recent and ongoing large-scale MT projects like the EU-funded TC-Star project and the US-DARPA-funded GALE project. In both projects, statistical MT is extended from text input to speech input.

Today, a typical state-of-the-art statistical MT system has the following four components:

1. Training: For each sentence pair in the training data, an alignment matrix is computed, typically by using the set of IBM-1 to IBM-5 alignment models and a Hidden Markov model.

2. Phrase extraction: From the alignment matrices of all training sentence pairs, source-target fragments are excised and used to define the so-called phrase tables.

3. Definition of the log-linear model: For each source-target phrase pair in the phrase table, so-called scoring functions are defined. Based on the training data, these scoring functions compute a probabilistic score of the hypothesis that the source fragment and the target fragment under consideration are translations of each other. These scoring functions are complemented with a word and/or phrase re-ordering model. All these scoring functions are combined in a so-called log-linear model. The weight of each scoring function is tuned for optimal translation quality or a related criterion.

4. Generation or search: For the given source sentence, the goal is to select the target sentence with the highest probabilistic score in the log-linear model. To

this purpose, the search algorithm has to generate and score hypotheses along various dimensions: unknown segmentation of the source sentence, unknown target phrases and unknown order of these phrases in the target sentence.

This talk will review the details of these components and the progress that the field has made so far and will also compare the statistical approach with example- and memory-based approaches.

Panel Session: Is MT in Crisis?

Moderated by:

Steven Krauwer
Professor of Computational Linguistics
ELSNET/University of Utrecht,
Trans 10,
NL-3512 JK Utrecht,
The Netherlands

steven@krauwer.nl

Some people maintain that MT is in deep crisis and has been so for many years. Others maintain that the crisis is just in the eye of the beholder and that MT is more flourishing than ever. Still others point to the fact that one man's (e.g. MT's) crisis is another man's (e.g. Translation Tools' or MT Evaluators') opportunity.

If you don't know the answer to this question you should attend the panel session, where five high-level experts will give you the real and ultimate answer. If you *do* know the answer you should attend to check whether the experts got it right, and to contradict them if necessary.

TMI-07

11th International Conference on Theoretical and Methodological Issues in Machine Translation

Table of Contents

An Assessment of Language Elicitation without the Supervision of a Linguist

Alison Alvarez, Lori Levin, Robert Frederking {[nosila|lsl|ref]@cs.cmu.edu}
Jill Lehman {jill@kidaccess.com}
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, Pennsylvania

ABSTRACT

The AVENUE machine translation system is designed for resource poor scenarios in which parallel corpora are not available. In this situation, parallel corpora are created by bilingual consultants who translate an *elicitation corpus* into their languages. We have described the elicitation corpus in other publications. This paper is concerned with evaluation of the elicitation corpus: is it suitably designed so that a bilingual consultant can produce reliable data without the supervision of a linguist? We evaluated two translations of the English elicitation corpus, one into Thai and one into Bengali. Two types of evaluation were conducted: an error analysis of the translations produced by the Thai and Bengali consultants, and a comparison of Example Based MT trained on the original translations and on corrected translations.

1 INTRODUCTION

MT systems can be learned from large parallel corpora or they can be produced by humans writing rules. A few researchers have investigated whether, in the absence of human rule writers and corpora, an MT system can be learned from linguistically naïve human consultants (McShane and Nirenburg, 2003, McShane et al. 2002; Probst, 2005). Two approaches have been taken. The Boas system (McShane et al, 2002) trains the consultants in linguistic terminology and then asks them whether their language has, for example, nominative case or dual number. Our work relies on having the consultant translate a list of sentences, or "elicitation corpus", that is like a fieldworker's questionnaire. Each sentence is designed to elicit a specific morphosyntactic property of the language. For example, we compare the translation of *A tree fell* and *Two trees fell* to see if verbs agree with subjects in number.

Our approach relies on the consultant getting the point of each example, with minimal use of linguistic terminology (see below). But this approach can easily fail to produce data that is useful for training an MT system. For example, the consultant may speak a language that does not normally use articles, but may feel compelled to translate the English words *the* and *a*, resulting in a corpus and that translation may not accurately reflect the normal syntax of his or her language.

As part of a U.S. government project called REFLEX, we produced an elicitation corpus of 3124 English sentences, which the Linguistic Data Consortium (LDC) is translating into a number of languages, beginning with Thai and Bengali.

This paper is concerned with an evaluation of our elicitation corpus. Two types of evaluation are provided. First, we provide an error analysis of two human translations of the elicitation corpus. Second, we compare an Example Based MT (EBMT) system trained on original human-produced translations and on

srcsent: We baked cookies.
context: We = 5 men;

((actor ((np-function fn-actor) (np-general-type pronoun-type)(np-person person-first)
(np-identifiability identifiable) (np-pronoun-exclusivity inclusivity-neutral)
np-number num-pl) (np-biological-gender bio-gender-male)(np-animacy anim-human)
(np-specificity specific)(np-pronoun-antecedent antecedent-not-specified) (np-distance
distance-neutral)))
(undergoer ((np-function fn-undergoer)(np-person person-third)(np-identifiability unidentifiable)
(np-number num-pl)(np-specificity non-specific)(np-animacy anim-inanimate)
(np-biological-gender bio-gender-n/a)(np-general-type common-noun-type)
(np-pronoun-exclusivity inclusivity-n/a)(np-pronoun-antecedent antecedent-n/a)
(np-distance distance-neutral)))
(c-polarity polarity-positive) (c-v-absolute-tense past) (c-v-lexical-aspect activity-accomplishment)(c-general-type declarative-clause)(c-my-causer-intentionality intentionality-n/a)(c-comparison-type comparison-n/a)…

Figure 1: A source language sentence, its context field and its abridged feature structure.

corrected translations in order to see the extent to which the errors of a linguistically naïve translator affect translation quality. We will conclude by discussing the implications of using linguistically naïve consultants as a resource for building MT systems.

2 Background

The AVENUE project has two related foci: building MT systems in low-resource scenarios, and making robust, hybrid MT systems using combinations of deep linguistic knowledge and statistical techniques. The hybrid system is a *statistical transfer* system (Lavie et al. 2004), which makes use of transfer rules as well as a statistical decoder. The rules can be written by hand, or learned automatically (Probst 2005). The AVENUE system also includes an EBMT system (Brown 1996), in order to use any pre-existing parallel texts that do happen to be available.

One hypothesis of the AVENUE work for low-resource scenarios is that MT systems can be learned from small amounts of data if the data is highly structured (Lavie et al. 2003). The elicitation corpus is therefore designed to produce highly structured data. Each sentence is designed to elicit a specific morphosyntactic property of the language, and sentences are organized into minimal pairs (e.g., *A tree is falling* and *A tree fell*) to compare the effects of changing one grammatical feature at a time. Probst (2005) describes automatic rule learning from elicited data.

A small sample of elicitation sentences is included in the list below. A more detailed description of the elicitation corpus can be found in Alvarez et al, (2006).

- Mary is writing a book for John.
- Who let him eat the sandwich?
- Who had the machine crush the car?
- They did not make the policeman run.
- Our brothers did not destroy files.
- He said that there is not a manual.
- The teacher who wrote a textbook left.
- The policeman chased the man who was a thief.
- Mary began to work.

Each sentence in the elicitation corpus is associated with a set of feature-value pairs, which represent the meaning elements that may be reflected in the

morphosyntax of the language. Figure 1 shows an example of an elicitation sentence and its feature structure.

As mentioned above, the elicitation corpus was translated into Thai and Bengali. The structural differences between Thai and Bengali make them excellent choices for our first elicitation corpus assessment. Bengali is a synthetic Indo-European language spoken in India and Bangladesh. It has rich system of tense and aspect. Thai is a highly analytic language with a complex pragmatic system and gender marking. It is the national language of Thailand and is a member of the Tai-Kadai language family.

In our analysis of corpus translations, we found 1064 elicitation errors in the Thai Corpus and 359 in the Bengali corpus. An elicitation error is any translation mistake that would lead to an incorrect characterization of a language. A discussion of these types of mistakes can be found in section 4.

We also wanted to see to what degree these translation errors in the corpus would harm an MT system learned from the data. For a variety of reasons, it was not practical to train our statistical transfer system on this data. We therefore assessed the impact of these elicitation errors by training two EBMT systems on our Thai data. One trained on our original unsupervised corpus and the other trained on a corpus corrected of elicitation errors. This evaluation is described in section 6.

3 Related Work

Two other projects that we know of formulate grammars based on elicited data. In addition to the Boas system mentioned above, which attempts to train naïve informants to provide linguistic information, the Grammar Matrix (Bender and Flickinger, 2005) collects facts like the existence of subject-verb agreement from a field worker and then automatically produces an HPSG grammar for the language. Both of these use knowledge that a trained human has put into technical linguistic form. In contrast, our approach analyzes translations of elicitation corpus sentences, and the underlying feature structures they represent, to derive the linguistic facts about the language automatically.

3 The Corpus and Support Materials

Our elicitation corpus is a monolingual corpus of 3124 English sentences. We designed it to be translated into any human language. Each sentence in the untranslated corpus is made of three main components. First, we start with a feature structure that represents the elements of meaning that will be in the elicitation sentence. This structure has separate fields each representing head-bearing phrases. Each field contains a list of features and values that represent the pieces of meaning underlying the source language sentence. By features we mean morphosyntactic phenomena, for example, person, number or tense (Alvarez et al 2006).

Next, we annotated each feature structure with an English sentence that would represent the features and values in its underlying structure. Because our feature structures are intended to cover the majority of morphosyntactic features that exist in human language, our English sentence may not adequately represent all of the features in the feature structure. For example, given the sentence “We baked cookies”, some languages would translate it differently based on whether the actor was dual, plural, male or female.

If a linguist were to administer this corpus it would be possible for the language consultant to ask clarification questions. However, for the REFLEX project, the LDC administered the translation of our corpus with a single translator per language and with no supervision from our team. We had no contact with the translators during translation of the elicitation corpus and were not present to answer questions. To clear up confusion about how we wanted

the corpus sentences to be translated we used "context fields". The context field supplements our English elicitation sentences with information not easily represented in the English sentence itself, but represented in the feature structure.

Our feature structures by themselves are complicated and would be difficult for someone without linguistic training to understand. However, a context field and a source sentence together embody all of the information in their corresponding feature structure. Thus, we were able to hide the feature structure and give the translators just the elicitation sentence and context.

2a.
Sentence: You wrote.
Context: You = five men
Translation: antum katabtum

2b.
Sentence: You wrote.
Context: You = two men
Translation: antumaa katabtumaa

2c.
Sentence: You wrote.
Context: You = five men
Translation: escribieron

2d.
Sentence: You wrote.
Context: You = two men
Translation: escribieron

Figure 2: Context information isn't always incorporated into target language translations. The two sentences translated into Modern Standard Arabic (2a and 2b) are translated differently based on the number of people 'You' represents. However, the Spanish translations remain the same in 2c and 2d. This example and further ones can be found in our translator guide (Alvarez et al. 2007).

For further clarification, we wrote a translator guide with examples and explanations to steer the native speakers toward translations that would reveal the language features of the target language.

When we talk about revealing language features, we mean the morphosyntactic characterization of a language. That is, we want to be able to learn how language features are grammaticalized in a target language or if they are manifested at all. In our case, we strove to get the most natural sounding translation that would let us learn about the features of a language. This means that not every feature will be translated into our target elicitation language. This is an acceptable outcome as it is just as important to know what features are *not* grammaticalized in a language as those that are. For example, a Spanish speaker would translate the plural second person pronoun the same whether 'you' represented 2 or 5 people. However, in Modern Standard Arabic the two sentences would translate differently depending on whether the pronoun represented 2 or 5 people. Thus, the context field may play into the translation of one language, but not into another. Because we designed our corpus to be used with any language a translator may be faced with, context fields will contain information that that may or may not be able to be utilized by the language consultant. One of the tasks of our translator guide was to help the translator learn where to draw this line. The next section will examine the extent to which the guide achieved this goal and the extent to which we were able to acquire successful translations.

4 Elicitation Corpus Translation Assessment

We assessed our translations using methods similar to those used by field linguists (Longacre 1964). That is, we analyzed sentences by comparing them to one another in order to pick out translation patterns. However, the consequences of unsupervised translation cut both ways for us. Thus, while the translator was unable to get clarification directly from us, we were unable to get clarification directly from the translator. A linguist in the field would be able to ask the language

Thai Elicitation Errors		
Source Sentence Over-Translation	845	79.41%
Context Over-Translation	57	5.35%
Under-translation	88	8.48%
Mistranslation	68	6.39%
Grammar Mistakes	6	0.19%
Total	1064	100%

Bengali Elicitation Errors		
Source Sentence Over-Translation	0	0.0%
Context Over-Translation	24	6.68%
Under-translation	5	1.39%
Mistranslation	76	21.17%
Grammar and Spelling Mistakes	254	70.75%
Total	359	100%

Figure 3: Total elicitation errors for the Thai and Bengali translations of the elicitation corpus.

consultant about the meaning of individual words and morphemes, but without this resource we were forced to compensate with dictionaries, grammars and language learning materials in order to confirm correct translations. In cases where we were unable to account for every in a sentence we consulted with local native speakers to assess the meaning of unknown phenomena.

Based on this analysis, we were able to assess all of our Thai and Bengali translations and keep track of elicitation errors. By our standards, most sentences were translated in a way that would make them useful as a resource for learning about a target language. However, some sentences contained constructions that diminished the utility of the translation and would provide spurious information about the grammaticalization of the target language. Below you will find a classification of these errors and their consequences. For full results of these error types for Bengali and Thai see the tables in figure 3.

4.1 Context Over-translation

The elicitation corpus's context fields are designed to provide additional information that may or may not be used as clarification when translating a sentence. Referring back to figure 2, the distinction between dual and plural pronouns causes a difference in translation for the Arabic translation, but not for the Spanish. The information in the context field is not incorporated because the Spanish translations would be the same whether 'You' referred to two, five or a hundred people. The distinction between dual and plural pronouns is Spanish is not grammaticalized. However, if the translator is determined to use the information in the context field it is possible for them to translate the sentences into the Spanish equivalent of 'You two wrote' or 'You five wrote', or even 'You two men wrote' and 'You five men wrote'. While grammatical, the excess information does not clarify the translation, and furthermore, it adds information not found in the source sentence. Thus, if the over-translated source and target sentences were to be fed to a word alignment system or a statistical machine translation system we would see 'You wrote' aligned with the Spanish equivalent of 'You two wrote'. This increases the chance of generating incorrect translations and will reduce the quality of the translation system.

Furthermore, this error type can lead to translations that are awkward. The goal of our corpus is to elicit translations as they exist in their target language naturally.

An example of this elicitation error can be found in (a) in figure 4. The Bengali instance over-translates the distant past tense. In Bengali, the simple past

a. Context Over-translation

Bengali target: বিজয়া কয়েক সপ্তাহ আগে বংকীমকে বইগুলি দিচ্ছিল.

transliteraton:	BAiJAYYAaa	KAYYAeKA	SAPAVIRTAaaHA	AAGAe
	BAANUKAiiMAKAe	BAIGAuLAi	DAiCAVIRCHAiLA.	
gloss:	Bijoya	a-few	moment-plural	before
	Bankim-acc	books-plural	give/third-person/progressive	

source: Bijoya was giving Bankim books.
context: Translate this sentence as if the incident it refers to happened minutes ago.

b. Source Sentence Over-translation

Thai target: ผู้ชาย คน นั้น ได้ มี ความสุข

transliteration:	pôo chaai	kon	nán	mee	kwaam sòok
gloss:	man	person	that	is	happy

srcsent: The man was happy.
context:

c. Under-translation

Thai target: ผู้ชาย คน นั้น จะ ตำหนิ เด็กผู้หญิง คน นั้น

Transliteration:	pôo chaai	kon	nán	jà	dtam-nì dèk	pôo ying	kon	nán
gloss:	man	person	that	will	reprimand	girl	person	that

srcsent: The man will criticize the girl.
context: Translate this as if the speaker heard this information from a rumor.

d. Mistranslation

Thai target: รั้ว รอบ ทุ่งหญ้า พังทลาย ลง

Transliteration:	rúa	rôp	tôong yâa	pang tá-laai	long
gloss:	fence	around	pasture	fall	down

srcsent: The fence around the pasture collapsed.
context:

e. Spelling and Grammar Mistakes

Bengali target: মহিলাটি যে গুদামে নয় কথা বলিতেছে.

Transliteration:	MAHiLaaTTi	Ye	GAuDAaaMAe	NAYYA	KATHAaa
	BALAiTAeCHAe.				
gloss:	woman-def	what	store	negative	statement
	talk/third-person/progressive				

srcsent: The woman who is not in the store is talking.
context:

Figure 4: This figure catalogs examples of our five types of elicitation errors. They are discussed in the text.

tense of an action remains the same whether it occurred seconds, days or years ago. The Bengali translation for sentence (a) now means 'Bijoya was giving Bankim books a few moments before.' if translated back into English. This translation does not match the meaning of the source sentence or its feature structure.

4.2 Source Sentence Over-translation

Source sentence over-translations occur when the translator over-specifies the translation in order to match the source sentence at the sacrifice of fluency or natural sounding translations. For example, in example b. found in figure 4 the Thai translator attempted to add definiteness to his/her translation by including the Thai demonstrative 'nán', which translates as 'that' in English.

There are two problems that arise

with this elicitation error. First, Thai doesn't mark definiteness explicitly, and certainly not with a demonstrative word. Secondly, the source and target language sentences have slightly different meanings. The original source sentence is '*The* man was happy,' but the translation means '*That* man was happy'. A more appropriate translation would have been 'pôo chaai kon mee kwaam sòok' or 'Man is happy'. While the ideal translation leaves the definiteness as ambiguous, it gives us a natural, reasonable translation, and, more importantly, gives us information about what features in the source sentence remain unmarked in the translation sentence.

Source sentence over-translation differs from context over-translation in one key way. In the case of source over-translation there is no information included in the target sentence that is not found in the source sentence. However, with context over-translation the target sentence includes information found in the source sentence that should remain unspecified in the translation. So, source sentence over-translations include too many features from the source and context over-translation includes too many from the context.

For the Thai elicitation corpus, source sentence over-translation was the most prevalent elicitation error found, but it is relatively rare in the Bengali corpus. This can be explained by how closely each language is related to English. Like English, Bengali is an Indo-European language. In addition it marks definiteness and number just as English does. However, Thai leaves both of these features unmarked morphosyntactically. In fact, out of the 845 Thai over-translation errors over 578 were made over specifying definiteness, identical mistakes that were repeated over and over again. This feature couldn't be over-translated in Bengali because it is marked morposyntactically just as in English. This explains the total of zero source sentence over-translations for Bengali.

4.3 Under-translation

Under-translation occurs when information from the context or source sentence is not translated into the target sentence. Thus, under-translation is an elicitation error caused by leaving something out. For example, substituting the word for 'person' for that of 'woman' or 'man' eliminates the feature of gender that would otherwise be evident in a sentence.

However, most under-translations are not that obvious. Under-translations can be difficult to find compared to over-translation. In our case, we discovered over-translations just by glossing sentences and double-checking those we discovered with a native speaker. In addition, we relied on language grammars and language typology charts (comparative tables indicating the morphosyntactic characteristics of many languages) to help discover this error.

The only under-translations we found were related to source marking. According to Iwasaki and Ingkaphirom (2005), evidentiality is marked in Thai analytically, especially in cases of hearsay. Our Thai translator, however, made no distinction between sentences describing events directly observed by the speaker and those heard from a rumor or gathered from evidence. Each sentence is translated grammatically, but omitting a key word that would give us insight into the categorization of information sources.

This elicitation error is rare, but having translators look at sentences within a narrative might mitigate this error, especially with regard to evidentiality.

4.4 Mistranslation

Mistranslations occur when the target sentence means something different from the source sentence. This means that the feature structure representing the meaning of the first sentence would be different than that of the target sentence

feature structure.

For example, one of the most common mistranslations involves mistaking the aspect represented by the source sentence. For example, a habitual source sentence might be translated as present progressive. Another example would be the Thai translation (d) in figure 4. A past tense English sentence was translated as a present tense Thai sentence. Thus the Thai translation would be translated back into English as 'The fence around the pasture collapses.' There is a natural, fluent way to translate the Thai sentence in the past tense, thus it is likely that the translator made a mistake and translated using the wrong tense.

One reason for the occurrence of this error might be that some of our English source sentences appear to be too ambiguous or have overly subtle distinctions. This might leave the translator to interpret the sentence to the best of his/her abilities and that interpretation might not match up with what we expect to elicit. Compounding this is the fact that some of our sentences are awkward, unclear or absent of a narrative. Of course, some of this may be attributed to human error. Out of several thousand sentences some mistakes can be expected.

4.5 Spelling and Grammar Mistakes

This elicitation error covers the spelling mistakes and grammar mistakes that happen within the corpus. Also included in this category are sentences that are faithful translations, but are ungrammatical in the target language. A certain degree of human error can be expected; the frequency of this type of mistake will depend on the education level of the translator.

However, large numbers of these elicitation errors could point to larger difficulties with translations. A portion of our Bengali elicitation corpus contains a number of recurring mistakes that are unlikely to have been made by a native speaker.

For example, the Bengali sentence (e) in figure 4 is an ungrammatical way to represent a relative clause in Bengali. In reality this sentence would have to be translated with two separate clauses which can be taken to mean the following as an English equivalent: 'The woman who is angry, she is talking'. It is possible that the translator was trying too hard to stick to the structure of the English translation, but the Bengali sentence as it stands is not correct Bengali in any dialect.

Further mistakes were made with regard to using inanimate markers on animate noun phrases and the use of classical Bengali in inappropriate contexts. The common Bengali name 'Bankim' was even spelled incorrectly for a portion of the corpus. Both of our native speaker consultants agreed that translations involving these mistakes were unlikely to have been made by a native speaker.

These mistakes were the most popular for the Bengali corpus and accounted for 254 total errors, or 70.75%. In comparison, the Thai corpus only contained a total of 6 spelling and grammar mistakes.

5 Suggestions for Improving the Elicitation Error Rate

The cause of these elicitation errors could come from three places.

First, our documentation may not be clear enough. It could be lacking in examples or be lacking in clarity. We were hindered because we were forced to use translation examples from an assortment of languages, none of which are the language of the translators, to illustrate our arguments. However, the translators seemed to have understood the documentation and followed its directions. They made few mistakes with regard to the context field and only over interpreted it in 57 out of 3124 sentences for Thai and 24 out of the same number for Bengali. Even the error of source over-translation,

while widespread, did not occur 100% of the time in places where it could have appeared. For Thai, it seems that our Thai translator was torn between delivering natural translations and delivering ones that conformed as closely as possible to the English source sentence. In light of this, we will be adding further examples to the documentation to clarify this, the most prevalent translation error.

Secondly, it is possible that some of the elicitation corpus sentences are unwieldy and difficult to translate. Magnifying this awkwardness is the fact that our sentences are without discourse context. That is, the sentences might benefit from appearing as part of a larger narrative or a story. Other sentences, such as those exploring locative features might benefit from pictures or other visual aids to clarify the meaning of each locative construction. Field linguists often use pictures or stories to clarify their elicitation sentences, so it might be of benefit to us to do the same.

Lastly, it is possible that our corpus is *too* unsupervised. A short period of training for the translators would be a way to catch and correct common types of elicitation errors. Though the point of this corpus is to perform unsupervised elicitation, it could be beneficial to administer a short pre-test with detailed feedback. This strategy could be a way to catch the most common elicitation mistakes. Our most common elicitation errors were really one mistake repeated many times. As we said in section 4.2, our Thai translator over-translated definiteness 578 times. Eliminating just this mistake reduces the elicitation error by 68.4%. Caught early, these easily correctable mistakes could dramatically improve our chances of getting the translations we desire.

6 Elicitation Errors and Machine Translation

To further assess the impact of elicitation errors found within unsupervised elicitation corpora, we trained two EBMT systems (Brown, 1996) to compare the results between one trained on our unsupervised data and one trained on the same data cleaned of elicitation errors. This corrected corpus will represent an ideal corpus translated under the supervision of a linguist.

Of the two corpora available, we chose to work with Thai rather than Bengali. This is because the errors for the Bengali corpus were too extensive to be corrected by a non-native speaker. Additionally, the errors in the Thai corpus were repetitive and less resource intensive to correct. Furthermore, the lack of morphology and the stable orthography made Thai the clear choice for a machine translation system trained on such a small corpus without segmentation.

We translated from Thai to English. The system trained only on about 2900 sentences from our elicitation corpus. The training sets used by our two EBMT systems used corresponding sentences for training data. This means that if a specific sentence from the uncorrected corpus were to be added to the training set, its corrected counterpart would be added to the set of training data for our corrected elicitation corpus.

Of the remaining 200 sentences, 100 were using for tuning the systems and 100 were used for testing. The test sentences in both cases were from the corrected corpus, since we want to test against gold standard translations. We also used a pre-trained English language model to aid in output generation.

Our results are displayed in the table below:

EBMT BLEU Results	
Uncorrected Thai	0.499
Corrected Thai	0.552

There is a 9.6% difference between the scores of the two systems. The Bleu scores are high due to the short sentences in our test set and the redundancy throughout our corpus.

Because we trained and tested only on the source and target sentences without their contexts there will be a number of sentences with duplicates in the corpus. Sentences that are found both in the training and target sets are assured perfect matches from the EBMT system and contributed to the high Bleu scores.

However, we are more interested in the difference between the two scores than in the performance of the systems themselves. The 9.6% difference is significant, but the uncorrected data system was still in a comparable range with the one trained on corrected data.

7 Conclusion

While there were numerous elicitation errors occurring with both the Thai and Bengali elicitation corpora, these errors were not so serious that they would render sentences useless for learning about a language, especially for human analyzers.

Elicitation errors also significantly affected the performance of the EBMT system. However, despite this, the Bleu score declined by less than 10%, providing some evidence that the uncorrected translations would still be able to train a usable system.

We will conduct further experiments to gauge the effect of elicitation errors on larger sets of training data. We will also investigate methods for recovering from noise in our training data, when it is not systematic.

8 Acknowledgements

We would like to thank our language consultants, Dipanjan Das, Satanjeev Bannerjee and Vorachai Tejapaibul. In addition, we would like to thank Aaron Phillips for his help training and testing our EBMT system.

9 References

Alvarez, Alison, Lori Levin, Robert Frederking. (2007) Elicitation Corpus Translator Guide. Technical Report. To appear.

Alvarez, Alison, Lori Levin, Robert Frederking, Simon Fung, Donna Gates and Jeff Good. (2006) "The MILE Corpus for Less Commonly Taught Languages". *Proceedings of HLT- NAACL-2006*, New York.

Bender, Emily M. and Dan Flickinger. 2005. Rapid Prototyping of Scalable Grammars: Towards Modularity in Extensions to a Language-Independent Core. *Proceedings of IJCNLP-05 (Posters/Demos)*, Jeju Island, Korea.

Brown, Ralf D. (1996) "Example-Based Machine Translation in the Pangloss System". In *Proceedings of the 16th International Conference on Computational Linguistics* (COLING-96), Denmark.

Iwasaki, Shoichi, Preeya Ingkaphirom. (2005) *A Reference Grammar of Thai*. Cambridge University Press.

Lavie, Alon, Stephan Vogel, Lori Levin, Erik Peterson, Katharina Probst, Ariadna Font Llitjos, Rachel Reynolds, Jaime Carbonell, and Richard Cohen. (2003) "Experiments with a Hindi-to-English Transfer-based MT System under a Miserly Data Scenario". *ACM Transactions on Asian Language Information Processing (TALIP), 2(2)*. June 2003. Pages 143-163.

Lavie, Alon, Shuly Wintner, Yaniv Eytani, Erik Peterson and Katharina Probst. "Rapid Prototyping of a Transfer-based Hebrew-to-English Machine Translation System". In Proceedings of TMI-2004, Baltimore, MD.

Longacre, Robert. (1964) *Grammar Discovery Procedures*. Mouton & Company, the Hague.

McShane, Marjorie, Sergei Nirenburg. (2003) Parameterizing and Eliciting "Text Elements across Languages for Use in Natural Language Processing Systems". *Machine Translation 18(2):* 129-165

McShane, Marjorie, Sergei Nirenburg, James Cowie, and Ron Zacharski. (2002) "Embedding knowledge elicitation and MT systems within a single architecture." *Machine Translation* 17(4).271–305.

Probst, Katharina. (2005) *Learning Transfer Rules for Machine Translation with Limited Data*. Ph D. Dissertation, Carnegie Mellon University.

Combining translation models in statistical machine translation

Jesús Andrés-Ferrer
PRHLT Group
UPV
jandres@dsic.upv.es

Ismael García-Varea
PRHLT Group
UCML
ivarea@info-ab.uclm.es

Francisco Casacuberta
PRHLT Group
UPV
fcn@dsic.upv.es

Abstract

Originally, statistical machine translation was based on the use of the "noisy channel" approach. However, many of the current and successful statistical machine translation systems are based on the use of a direct translation model or even on the use of a log-linear combination of serveral direct and inverse translation models. An attempt to justify the use of these heuristic systems was proposed within the framework of maximum entropy.

We present a theoretical justification under the decision theory framework. This theoretical frame entails new methods for increasing the performance of the systems combining translation models. We propose new and more powerful translation rules that also fit within this theoretical framework. The most important theoretical properties developed in the paper are experimentally studied through a simple translation task.

1 Introduction

Machine Translation (MT) deals with the problem of automatically translating a sentence ($\mathbf{f}$) from a source language[1] ($\mathbf{F}^*$) into a sentence ($\mathbf{e}$) from a target language ($\mathbf{E}^*$). Obviously, these two languages are supposed to have a very complex set of rules involved in the translation process that cannot be properly enumerated into a computer system. According to this, many authors have embraced a statistical approach to the MT problem, where the only source of information is a parallel corpus of source-to-target translated sentences.

Brown et al. (1993) approached the problem of MT from a purely statistical point of view. In this approach, the MT problem is analysed as a classical pattern recognition problem using the well-known Bayes' classification rule (Duda et al., 2000). Therefore, statistical machine translation (SMT) is a classification task where the set of classes is the set of all sentences of the target language ($\mathbf{E}^*$), i.e. every target string ($\mathbf{e} \in \mathbf{E}^*$) is regarded as a possible translation for the source language string ($\mathbf{f}$). The goal of the translation process in statistical machine translation can be formulated as follows: a source language string $\mathbf{f}$ is to be translated into a target language string $\mathbf{e}$[2]. Then the system searches the target string ($\hat{\mathbf{e}}$) with maximum a-posteriori probability $p(\mathbf{e}|\mathbf{f})$:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*}\{p(\mathbf{e}|\mathbf{f})\} \qquad (1)$$

where $p(\mathbf{e}|\mathbf{f})$ can be approached through a *direct* statistical translation model. Eq. (1) has proved to be the optimal

[1] $\mathbf{F}^*$ is the set of all possible strings with a finite length on the lexicon $\mathbf{F}$.

[2] We will refer to $p(\mathbf{e}|\mathbf{f})$ as a direct statistical translation model and to $p(\mathbf{f}|\mathbf{e})$ as an inverse statistical translation model.

decision/classification rule under some assumptions and is called the optimal Bayes' classification rule (obviously assumes that the actual probability distribution $p(\mathbf{e}|\mathbf{f})$ is known). Applying the Bayes' theorem to Eq. (1), the following rule is obtained:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*}\{p(\mathbf{e})\cdot p(\mathbf{f}|\mathbf{e})\} \qquad (2)$$

Eq. (2) implies that the system has to search the target string ($\hat{\mathbf{e}}$) that maximises the product of both, the target language model $p(\mathbf{e})$ and the inverse string translation model $p(\mathbf{f}|\mathbf{e})$. Thus, the Bayes' classification rule provides the *inverse translation rule* (ITR), which is also called "the fundamental equation of SMT". Again, this rule is optimal if the actual models are known. Nevertheless, using this rule implies, in practice, changing the distribution probabilities as well as the models through which the probabilities are approached. This is exactly the advantage of this approach, as it allows the modelling of the direct translation probability ($p(\mathbf{e}|\mathbf{f})$) with two models: an inverse translation model that approximates $p(\mathbf{f}|\mathbf{e})$; and a language model that approximates $p(\mathbf{e})$.

This approach has a strong practical drawback: the search problem[3]. This search is known to be an NP-hard problem (Knight, 1999; Udupa and Maji, 2006). However, several search algorithms have been proposed in the literature to solve this ill-posed problem efficiently (Brown and others, 1990; Wang and Waibel, 1997; Yaser and others, 1999; Germann and others, 2001; Jelinek, 1969; García-Varea and Casacuberta, 2001; Tillmann and Ney, 2003).

In order to alleviate this drawback, many of the current SMT systems (Och et al., 1999; Och and Ney, 2004; Koehn et al., 2003; Zens et al., 2002) have proposed the use of the *direct translation rule* (DTR):

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*}\{p(\mathbf{e})\cdot p(\mathbf{e}|\mathbf{f})\} \qquad (3)$$

which can be seen as an heuristic version of the ITR (Eq. (2)), where $p(\mathbf{f}|\mathbf{e})$ is substituted by $p(\mathbf{e}|\mathbf{f})$. This rule allows an easier search algorithm for some of the translation models.

Although the DTR has been widely used, its statistical theoretical foundation has not been clear for long time, as it seemed to be against the Bayes' classification rule if an *asymmetric model*[4] is used for modelling the translation probability. Other authors (Andrés-Ferrer et al., 2007) have provided an explanation of its use within decision theory. In this work, we expand that theory to other translation models and other loss functions, providing a general framework to combine translation systems.

Some of the current SMT systems (Och and Ney, 2004; Marino et al., 2006) use a log-linear combination of statistical models to approximate the direct translation distribution:

$$p(\mathbf{e}|\mathbf{f}) \approx \frac{\exp\left[\sum_{m=1}^{M}\lambda_m h_m(\mathbf{f},\mathbf{e})\right]}{\sum_{\mathbf{e}'}\exp\left[\sum_{m=1}^{M}\lambda_m h_m(\mathbf{f},\mathbf{e}')\right]} \qquad (4)$$

where h_m is a logarithmic statistical model that approximates a probability distribution (i.e. translation or language probabilities).

The paper is organised as follows: section 2 summarises the Bayes' decision theory. Section 3 tackles SMT under the decision theory framework. Finally, section 4 demonstrates in practice the theoretical ideas explained in the paper. Conclusions are condensed in the section 5.

2 Bayes Decision Theory

A classification problem such as the SMT problem can be seen as an instance of a Decision Problem (DP). From this point of view, a classification problem is composed of three different items:

1. A set of *Objects* ($\mathcal{X}$) the system might observe and has to classify (i.e., translate).
2. A set of classes ($\Omega = \{\omega_1,\ldots,\omega_C\}$) in which the system has to classify each observed object $\mathbf{x}\in\mathcal{X}$.

[3] The method for solving the maximisation (or the search) of the optimal $\hat{\mathbf{e}}$ in the set $\mathbf{E}^*$, i.e. $\arg\max_{\mathbf{e}\in\mathbf{E}^*}$

[4] Given two sentences $\mathbf{e}$ and $\mathbf{f}$ from the target and source language: a *symmetric* model assigns the same probability to $p(\mathbf{e}|\mathbf{f})$ and to $p(\mathbf{f}|\mathbf{e})$; and an *asymmetric* model does not.

3. A *Loss function* ($\mathrm{l}(\omega_k|\mathbf{x},\omega_j)$). This function evaluates the loss of classifying an observed object $\mathbf{x}$ in a class, $\omega_k \in \Omega$, knowing that the *optimal class* for the object $\mathbf{x}$ is $\omega_j \in \Omega$.

Therefore, when an object $x \in \mathcal{X}$ is observed in a classification system, the system chooses the "correct" class from all possible classes (Ω). The term "correct" is used in the sense of the action that minimises the loss in which the system could incur if it makes an error, according to the loss function. For reasons of simplicity, the 0-1 *loss function* is usually assumed, i.e.:

$$\mathrm{l}(\omega_k|\mathbf{x},\omega_j) = \begin{cases} 0 & \omega_k = \omega_j \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

This loss function does not penalise the correct class, nevertheless it does not distinguish between the importance of classifying an object in a specific wrong class or in another wrong class. Therefore, the penalty of classifying the object $\mathbf{x}$ in the class ω_i or ω_j is the same. This is only sensible in some small and simple cases. For example, if the set of classes is large, or even infinite (but still enumerable), then it is not very appropiate to penalise all wrong classes the same. Note that in this case it is impossible to define a uniform distribution over the classes. This implies that there are classes that have a very small probability, and then it does not make sense to define a uniform loss function for those classes. Instead, it is better to penalise the zones where the probability is high.

In order to build a classification system the *classification function* must be defined, say $\mathrm{c} : \mathcal{X} \to \Omega$. The class provided by the classification function may not be the correct class. Thereby, the classification function yields an error or risk, the so-called *Global Risk*,

$$R(\mathrm{c}) = \mathrm{E}_{\mathbf{x}}[R(\mathrm{c}(\mathbf{x})|\mathbf{x})] = \int_{\mathcal{X}} R(\mathrm{c}(\mathbf{x})|\mathbf{x})\,\mathrm{p}(\mathbf{x})d\mathbf{x} \tag{6}$$

where $R(\omega_k|\mathbf{x})$ (with $\omega_k = \mathrm{c}(\mathbf{x})$) is the *Conditional Risk given* $\mathbf{x}$, i.e. the expected loss of classifying in the class determined by the decision function. This *Conditional Risk* is expressed as follows:

$$R(\omega_k|\mathbf{x}) = \sum_{\omega_j \in \Omega} \mathrm{l}(\omega_k|\mathbf{x},\omega_j)\,\mathrm{p}(\omega_j|\mathbf{x}) \tag{7}$$

The well-known *Bayes' classification rule* is the rule that minimises the Global Risk. Moreover, as minimising the Conditional Risk for each object ($\mathbf{x}$) is a sufficient condition to minimise the Global Risk, without loss of generality we can say that the optimal *Bayes classification rule* is the rule that minimises the Conditional Risk, i.e.:

$$\hat{\mathrm{c}}(\mathbf{x}) = \arg\min_{\omega \in \Omega} R(\omega|\mathbf{x}) \tag{8}$$

Loss functions that are more appropriate than the 0-1 can be designed. If we only assume that the loss of correctly classifying an object is 0, then a very general loss function is obtained:

$$\mathrm{l}(\omega_k|\mathbf{x},\omega_j) = \begin{cases} 0 & \omega_k = \omega_j \\ \epsilon(\mathbf{x},\omega_k,\omega_j) & \text{otherwise} \end{cases} \tag{9}$$

In the case of Eq.(9), the optimal Bayes' classifier is given by:

$$\hat{\mathrm{c}}(\mathbf{x}) = \arg\min_{\omega_k \in \Omega} \sum_{\omega_j \neq \omega_k} \epsilon(\mathbf{x},\omega_k,\omega_j)\,\mathrm{p}(\omega_j|\mathbf{x}) \tag{10}$$

Note that in order to perform the search for the optimal class $\hat{\mathrm{c}}(\mathbf{x})$ it is necessary to find the class ω_k, for which the sum over all the remaining classes ω_j is mimimun. This requires a computation time[5] of $O(|\Omega|^2)$. This cost can be prohibitive in some problems. For instance, in machine translation, the set of classes is exponential with the length of the sentence. In this case, having to compute the sum for each class is a practical problem that can ruin the advantages obtained by using a more appropriate loss function.

In this sense, there is a particular set of loss functions of the form of Eq. (9), that preserves the simplicity of the optimal classification rule for the 0-1 loss function. If ω_k is the class proposed by the system and ω_j is the correct class

[5] Note that we are assuming that the cost of evaluating $\epsilon(\mathbf{x},\omega_k,\omega_j)$ and $\mathrm{p}(\omega_j|\mathbf{x})$ is costant in time

that the system should choose (ω_k is expected to be equal to ω_j) the following loss function $\mathrm{l}(\omega_k|\mathbf{x},\omega_j)$ preserves this simplicity:

$$\mathrm{l}(\omega_k|\mathbf{x},\omega_j) = \begin{cases} 0 & \omega_k = \omega_j \\ \epsilon(\mathbf{x},\omega_j) & \text{otherwise} \end{cases} \quad (11)$$

where $\epsilon(\cdot)$ is a function depending on the object ($\mathbf{x}$) and the correct class (ω_j) but not depending on the wrong class proposed by the system (ω_k). This function must verify that $\sum_{\omega_j \in \Omega} \mathrm{p}(\omega_j|\mathbf{x})\,\epsilon(\mathbf{x},\omega_j) < \infty$; and it evaluates the loss function when the system fails.

In such cases, it can be easily proved that the Conditional Expected Risk is:

$$R(\omega_k|\mathbf{x}) = \mathrm{S}(\mathbf{x}) - \mathrm{p}(\omega_k|\mathbf{x})\,\epsilon(\mathbf{x},\omega_k) \quad (12)$$

where $\mathrm{S}(\mathbf{x}) = \sum_{\omega_j \in \Omega} \mathrm{p}(\omega_j|\mathbf{x})\,\epsilon(\mathbf{x},\omega_j)$ and $\mathrm{S}(\mathbf{x}) < \infty$, i.e. the weighted sum over all possible classes converges to a finite number which only depends on $\mathbf{x}$. Therefore, $\epsilon(\cdot)$ is restricted to functions that hold the previous finiteness property.

As a result, the classification rule is very similar to the optimal Bayes' classification rule for the 0-1 loss function and simplifies to the following equation (Andrés-Ferrer et al., 2007):

$$\hat{\mathrm{c}}(\mathbf{x}) = \arg\max_{\omega \in \Omega} \{\mathrm{p}(\omega|\mathbf{x})\,\epsilon(\mathbf{x},\omega)\} \quad (13)$$

It is worth noting that the computational time[6] needed to sovle the search of the optimal class in Eq. (13). is $O(|\Omega|)$.

In conclusion, for each loss function there exists a different optimal Bayes' classification rule, specifically using a loss function like the one in Eq. (11) yields one of the simplest optimal classification rules, Eq. (13).

3 Statistical Machine Translation

SMT is a specific instance of a classification problem where the set of possible classes is the set of all the possible sentences that might be written in a target language, i.e. $\Omega = \mathbf{E}^*$. Likewise, the objects to be classified[7] are sentences of a source language, i.e. $\mathbf{f} \in \mathbf{F}^*$.

In a SMT system, the Bayes' classification rule is Eq. (2). As stated above, this classification rule can be obtained by using the 0-1 loss function:

$$\hat{\mathbf{e}} = \hat{\mathrm{c}}(\mathbf{f}) = \arg\max_{\omega_k \in \Omega} \{\mathrm{p}(\omega_k|\mathbf{f})\} \quad (14)$$

where $\omega_k = \mathbf{e}_k$. This loss function is not particularly appropriate when the number of classes is huge as occurs in SMT problems. Specifically, if the correct translation for the source sentence $\mathbf{f}$ is $\mathbf{e}_j$, and the hypothesis of the translation system is $\mathbf{e}_k$; using the 0-1 loss function (Eq. (5)) has the consequence of penalising the system in the same way, independently of which translation ($\mathbf{e}_k$) the system proposes and which is the correct translation ($\mathbf{e}_j$) for the source sentence ($\mathbf{f}$).

3.1 Quadratic loss functions

Equation (9) produces search algorithms which have a quadratic cost depending on the size of the set of classes. As stated above, machine translation can be understood as a classification problem with a huge set of classes. Hence, these loss functions yield difficult search algorithms. There are some works that already have explored this kind of loss functions (Ueffing and Ney, 2004; R. Schlüter and Ney, 2005).

The more appealing application of this loss functions is the use of a metric loss function (R. Schlüter and Ney, 2005). For instance, in machine translation one widespread metric is the WER (see Section 4 for a definition), since the loss function in Equation (9) depends on both, the proposed translation and the reference translation, the WER can be used as loss function (Ueffing and Ney, 2004). Nevertheless, due to the high complexity, the use of these quadratic and interesting loss functions, is only feasible in constrained situations like n-best lists (Kumar and Byrne, 2004).

[6] Note that we are assuming that the cost of evaluating $\epsilon(\mathbf{x},\omega_j)$ and $\mathrm{p}(\omega_j|\mathbf{x})$ is costant in time

[7] In this context to classify an object $\mathbf{f}$ in the class ω_k is a way of expressing that $\mathbf{e}_k$ is the translation of $\mathbf{f}$.

Another interesting loss function would be the one obtained by introducing a kernel as the loss function in Equation (9):

$$\mathrm{l}(\mathbf{e}_k|\mathbf{f},\mathbf{e}_j) = \begin{cases} 0 & \mathbf{e}_k = \mathbf{e}_j \\ \mathcal{K}_n(\mathbf{e}_k,\mathbf{e}_j) & \text{otherwise} \end{cases} \quad (15)$$

with

$$\mathcal{K}_n(\mathbf{e}_k,\mathbf{e}_j) = \sum_{\mathbf{u}\in E^n} |\mathbf{e}_j|_{\mathbf{u}}|\mathbf{e}_k|_{\mathbf{u}} \quad (16)$$

where $|\mathbf{e}|_{\mathbf{u}}$ stands for the number of occurrences of the sequence of n words $\mathbf{u}$ inside the sentence $\mathbf{e}$ (Cortes et al., 2005).

3.2 Linear loss function

Equation (11) produces search algorithms which have a linear cost depending on the size of the set of classes. For instance, a more suitable loss function than the 0–1 loss, can be obtained using Eq. (11) with $\epsilon(\mathbf{f},\mathbf{e}_j) = p(\mathbf{e}_j)$:

$$\mathrm{l}(\mathbf{e}_k|\mathbf{f},\mathbf{e}_j) = \begin{cases} 0 & \mathbf{e}_k = \mathbf{e}_j \\ p(\mathbf{e}_j) & \text{otherwise} \end{cases} \quad (17)$$

This loss function seems to be more appropriate than the 0-1. This is due to the fact that if the system makes an error translating a set of source sentences, this loss function tries to force the system to fail in the source sentence ($\mathbf{f}$) whose correct translation[8] ($\mathbf{e}_j$) is one of the least probable in the target language. Thus, the system will fail in the least probable translations, whenever it gets confused; and therefore, the *Global Risk* will be reduced.

In addition, it is easy to prove (using Eq. (13)) that this loss function leads to the Direct Translation Rule in Eq. (3). Then, the DTR should work better than the ITR, from a theoretical point of view.

Nevertheless, the statistical approximations employed for modelling translation probabilities might not be symmetric, as is the case with IBM Models (Brown and other, 1993). Thus, the model error, could be more important than the advantage obtained from the use of a more appropriate loss function. Therefore, it seems a good idea to use the direct rule in the equivalent inverse manner so that the translation system will be the same and then these asymmetries will be reduced. By simply applying the Bayes' theorem to Eq. (3), we obtain the equivalent rule:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*} \left\{ p(\mathbf{e})^2 p(\mathbf{f}|\mathbf{e}) \right\} \quad (18)$$

The difference between the Eq (3) and Eq (18) can be used to measure the asymmetries of the translation models.

An alternative function to the proposed in Eq (17) is the loss function in Eq. (11) with $\epsilon(\mathbf{f},\mathbf{e}_j) = p(\mathbf{f},\mathbf{e}_j)$:

$$\mathrm{l}(\mathbf{e}_k|\mathbf{f},\mathbf{e}_j) = \begin{cases} 0 & \mathbf{e}_k = \mathbf{e}_j \\ p(\mathbf{f},\mathbf{e}_j) & \text{otherwise} \end{cases} \quad (19)$$

which leads to:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*} \{ p(\mathbf{f},\mathbf{e}) p(\mathbf{e}|\mathbf{f}) \} \quad (20)$$

Equation (20) is able to provide several optimal classification rules depending on which approximation is used to model the joint probaility ($p(\mathbf{f},\mathbf{e})$). The most important rule produced by this function is the *Inverse and Direct translation rule (I&DTR)*, which is expressed by the following equation:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*} \{ p(\mathbf{e}) p(\mathbf{f}\,|\,\mathbf{e}) p(\mathbf{e}\,|\,\mathbf{f}) \} \quad (21)$$

The interpretation of this rule is a refinement of the direct translation rule. In this case, if the system makes a mistake it is done in the least probable pairs ($\mathbf{f},\mathbf{e}$) in terms of $p(\mathbf{e},\mathbf{f})$.

More interesting loss functions can be obtained using information theory. For instance, we can penalise the system by the *remaining information*. That is, if we knew $p(\mathbf{e})$, then the information associated with a target sentence $\mathbf{e}_j$ would be $-\log(p(\mathbf{e}_j))$. The remaining information, or the information that the system has learnt when it fails is given by $-\log(1-p(\mathbf{e}_j))$. Hence, the system can be penalised with this score:

$$\mathrm{l}(\mathbf{e}_k|\mathbf{f},\mathbf{e}_j) = \begin{cases} 0 & \mathbf{e}_k = \mathbf{e}_j \\ -\log(1-p(\mathbf{f},\mathbf{e}_j)) & \text{otherwise} \end{cases} \quad (22)$$

[8] Here lies the importance of distinguishing between the translation proposed by the system ($\mathbf{e}_k$) and the correct translation ($\mathbf{e}_j$) of the source sentence($\mathbf{f}$).

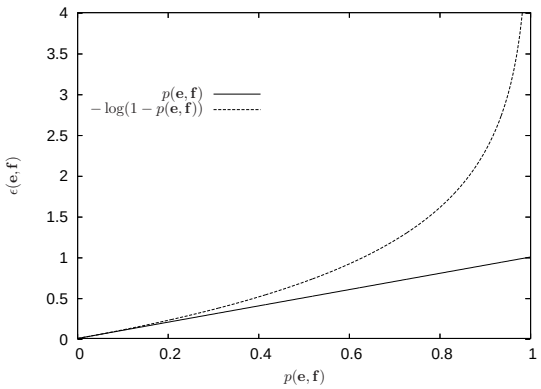


Figure 1: The information of the contrary event, or the remaining information.

Figure 1, shows the remaining information of a probability function. Note that the remaining information has a singularity at 1, i.e. if the system has not been able to learn a sure event, which has probability of 1, then the loss is infinity. Note that this loss can be defined for any probability such as $\mathrm{p}(\mathbf{e})$ or $\mathrm{p}(\mathbf{x},\mathbf{e})$.

Some works (Och and Ney, 2004; Marino et al., 2006), explore the idea of using maximum entropy models to design a translation system, obtaining in this way a translation rule of the form of:

$$\hat{\mathbf{e}} = \arg\max_{\mathbf{e}\in\mathbf{E}^*} \sum_{m=1}^{M} \lambda_m h_m(\mathbf{f},\mathbf{e}) \qquad (23)$$

where h_m is a logarithmic statistical model that approximates a probability distribution (i.e. translation or language probabilities).

The Eq (23) can be analysed from a Bayes' decision theory frame. Into this scope, what the log-linear systems are doing is to use the loss function in Eq (11) with:

$$\epsilon(\mathbf{f},\mathbf{e}) = p(\mathbf{e}\,|\,\mathbf{f})^{-1} \prod_{m=1}^{M} f_m(\mathbf{f},\mathbf{e})^{\lambda_i} \qquad (24)$$

where $f_m(\mathbf{f},\mathbf{e}) = \exp[h_m(\mathbf{f},\mathbf{e})]$.

From the decision theory, the log-linear models learn the best loss function among a family of loss functions. This family is defined by a vector of hyperparameters ($\boldsymbol{\lambda}_1^{\boldsymbol{M}}$):

$$\left\{ p(\mathbf{e}\,|\,\mathbf{f})^{-1} \prod_{m=1}^{M} f_m(\mathbf{f},\mathbf{e})^{\lambda_i} \;\middle|\; \forall\lambda_i \right\} \qquad (25)$$

In order to perform the optimisation, firstly the f_m functions (usually an exponential functions of probability distributions) are estimated using maximum likelihood (or some other estimation technique). Secondly, the ME algorithm (Berger et al., 1996) is used to find the optimal weights or hyperparameters λ_i, i.e., the ME algorithm is used to find the optimal loss function among all the possible functions in the family.

Some works explore the idea of using these hyperparameters to reduce the evaluation error metric, such as the Bleu (Papineni et al., 2001). For instance, in Och (2003), some improvements were reported when estimating the hyperparameters $\boldsymbol{\lambda}$ in accordance with the evaluation metric.

4 Experimental Results

The aim of this section is to demonstrate with practical results, how to use the theory stated in the work to improve the performance of a translation system. Obtaining a state-of-art system is out of scope of this paper. In this way, the previously stated properties will be analysed in practice with a simple translation model. In other works, some of the loss functions presented here has been analysed using state-of-art models, phrase-based models, (Andrés-Ferrer et al., 2007)

Before starting the section we need to define two new concepts (Germann and others, 2001). When a SMT system proposes a wrong translation, this is due to two reasons: the suboptimal search algorithm which has not been able to compose a good translation; or the model which is not able to make up a good translation (and so is unable to find it). Then we will say that a translation error is a *search error (SE)* if the probability of the proposed translations is less than the reference translation; otherwise we will say that it is a *model error*, i.e. if the probability of the proposed translations is greater than the reference translation.

We use the IBM Model 2 (Brown and other, 1993) and the corresponding search algorithms to design the experiments of this work. That choice was motivated by several

reason. Firstly, the simplicity of the translation model allows to obtain a good estimation of the model parameters. Secondly, there are several models that are initialised using the alignments and dictionaries of the IBM model 2. Finally, the search problem can be solved exactly using dynamic programming for the DTR.

In order to train the IBM Model 2 we used the standard tool *GIZA++* (Och, 2000). We re-implemented the algorithm presented in (García-Varea and Casacuberta, 2001) to perform the search process in translation for the ITR. Even though this search algorithm is not optimal, we set the parameters to minimise the search errors, so that all the errors should be model errors. In addition we implemented the corresponding version of this algorithm for the DTR and for the I&DTR. All these algorithms were developed by dynamic programming. For the I&DTR, we implemented two versions of the search: one guided by the direct model (a non-optimal search algorithm, namely I&DTR-D) and the other guided by the inverse translation model (which is also non-optimal but more accurate, namely I&DTR-I). Due to the length constraint of the article, the details of the algorithms are omitted.

We selected the Spanish-English TOURIST task (Amengual et al., 1996) to carry out the experiments reported here. The Spanish-English sentence pairs correspond to human-to-human communication situations at the front-desk of a hotel which were semi-automatically produced. The parallel corpus consisted of 171,352 different sentence pairs, where 1K sentences were randomly selected from testing, and the rest (in sets of exponentially increasing sizes: 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K and 170K sentences pairs) for training. The basic statistics of this corpus are shown in Table 1. All the figures show the confidence interval at 95%.

In order to evaluate the translation quality, we used the following well-known automatically computable measures:

1. *Word Error Rate* (WER):Word Error Rate is the minimum number (in %) of deletions, insertions, and substitutions that are necessary to transform the translation proposed by the system into the reference translation.

2. *Sentence Error Rate* (SER): Sentence Error Rate is the number (in %) of sentences that differs from the reference translations.

3. *BiLingual Evaluation Understudy* (BLEU): it is based on the n-grams of the hypothesized translation that occur in the reference translations. In this work, only one reference translation per sentence was used. The BLEU metric ranges from 0.0 (worst score) to 1.0 (best score) (Papineni et al., 2001):

	Test Set		Train Set	
	Spa	Eng	Spa	Eng
sentences	1K		170K	
avg. length	12.7	12.6	12.9	13.0
vocabulary	518	393	688	514
singletons	107	90	12	7
perplexity	3.62	2.95	3.50	2.89

Table 1: Basic statistics of the Spanish-English TOURIST task.

Figure 2 shows the differences in terms of the WER among all the mentioned forms of the DTR: "IFDTR" (Eq. 18), "DTR" (Eq. 3), and "DTR-N" (Normalised Length version of DTR). Note the importance of the model asymmetry in the obtained results. The best results were the ones obtained using the inverse form of the DTR. The normalised version was developed due to the fact that the IBM Model 2 (in its direct version) tries to provide very short translations. This behaviour is not surprising, since the only mechanism that the IBM Model 2 has to ensure that all sources words are translated is the length distribution. The length distribution usually allows the model to ommit the translation of a few words. Nevertheless, the "DTR" and "DTR-N" performed worse than the ITR (Table 2).

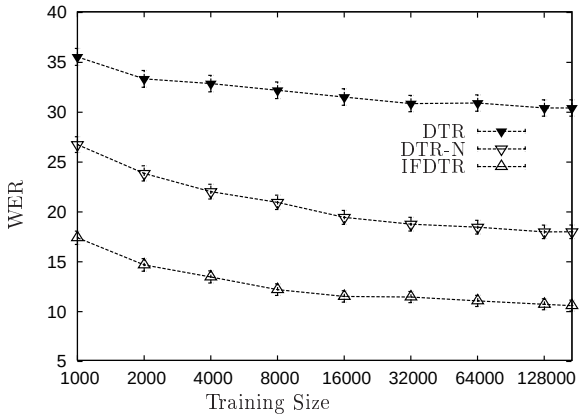


Figure 2: Asymmetry of the IBM Model 2 measured with the respect to the WER for the TOURIST test set for different training sizes.

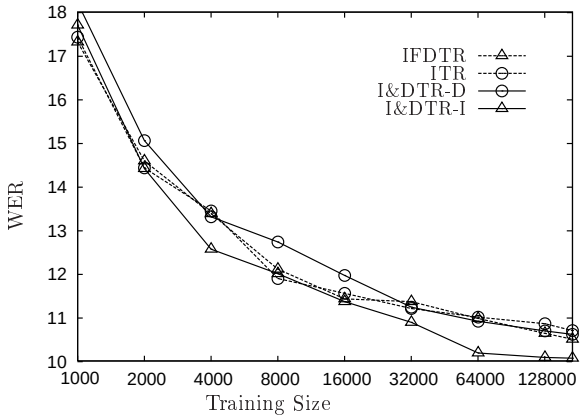


Figure 3: WER results for the TOURIST test set for different training sizes and different classification rules.

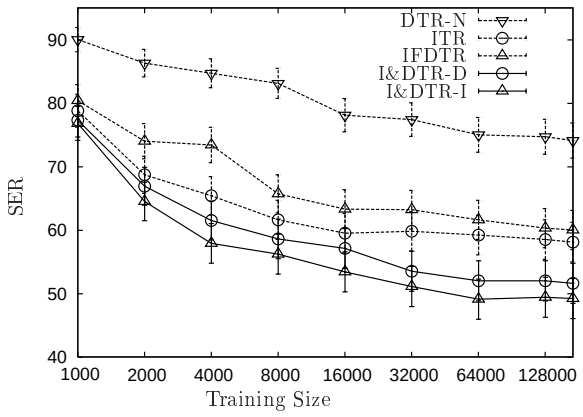


Figure 4: SER results for the TOURIST test set for different training sizes and different classification rules.

Model	WER	SER	BLEU	SE	T
I&DTR I	**10.0**	**49.2**	**0.847**	1.3	34
I&DTR D	10.6	51.6	0.844	9.7	2
IFDTR	10.5	60.0	0.837	2.7	35
ITR	10.7	58.1	0.843	1.9	43
DTR N	17.9	74.1	0.750	0.0	2
DTR	30.3	92.4	0.535	0.0	2

Table 2: Translation quality results with different translation rules for TOURIST test set for a training set of 170K sentences. Where T is the time expressed in seconds.

Figure 3 shows the results achieved with the most important rules. All the I&DTR obtain similar results to the ITR. Nevertheless, the non-optimal search algorithm guided by the direct model ("I&DTR-D") was an order of magnitude faster than the more accurate one ("I&DTR-I") and the ITR. The inverse form of the DTR ("IFDTR") behaved similarly to these, however improve the results reported by DTR. Therefore, there are no significant differences between the rules analysed in terms of WER. However, the execution times were significantly reduced by the direct guided search in comparison with the other searches. Table 2 shows these execution times and the figures with the maximum training size. Although the different search algorithms (based on loss functions) do not convey a significant improvement in WER. Note that the loss function only evaluates the SER, i.e. the loss function minimises the SER, and does not try to minimise the WER. Thus, changing the loss function, does not necessarily decrease the WER.

In order to support this idea, Figure 4 shows the analogous version of Figure 3 but with SER instead of WER. It should be noted that as the training size increases, there is a difference in the behaviour between the ITR and both I&DTR. Consequently, the use of these rules provides better SER, and this difference becomes statistically significant as the estimation of the parameters becomes better. In the case of the inverse form of the DTR ("IFDTR"), as the training size in-

creases, the error tends to decrease and approximate the ITR error. However, the differences are not statistically significant and both methods are equivalent from this point of view.

In conclusion, there are two sets of rules: the first set is made up of IFDTR and ITR, and the second is composed by the two versions of the I&DTR. The first set reports worse SER than the the second set. However, the I&DTR guided with the direct model ("I&DTR-D") has many good properties in practice.

5 Conclusions

The analysis of the loss function is an appealing issue. The results of analysing different loss functions range from allowing to use metric loss functions such as BLEU, or WER; to proving the properties of some outstanding classification rules such as the direct translation rule, the inverse translation rule or even the maximumn entropy rule. For each different function $\epsilon(\mathbf{f}, \mathbf{e}_j, \mathbf{e}_k)$ in the general loss function of Eq. (9), there is a different optimal Bayes' rule. The point of using one specific rule is an heuristic and practical issue.

An interesting focus of study is the use of metrics such as BLEU, or WER; as the loss function. Nevertheless due to the high complexity, it is only feasible on constrained situations like n-best lists.

This work focuses on the study of loss functions that have a linear complexity and that are outstanding due to historical or practical reasons. In this sense, we have provided a theoretical approach based on decision theory which explains the differences and resemblances between the Direct and the Inverse Translation rules. This theoretical frame predicts an improvement (in terms of SER), an improvement that has been confirmed in practice.

In order to increase performance, we should find the best loss function with the form in Eq (9) or with the form in Eq (11). As future work, we will develop this idea into detail under the scope of functional optimisation. We also intend to analyse the practical behaviour of other loss functions such as the loss functions in Eq.(15) or the *remaining information* loss function.

Acknowledgements

This work has been supported by the EC (FEDER), the Spanish MEC under grant TIN2006-15694-CO2-01 and the Valencian "Conselleria d'Empresa, Universitat i Ciència" under grant CTBPRA/2005/004.

References

J.C. Amengual, J.M. Benedí, M.A. Castaño, A. Marzal, F. Prat, E. Vidal, J.M. Vilar, C. Delogu, A. di Carlo, H. Ney, and S. Vogel. 1996. Definition of a machine translation task and generation of corpora. Technical report d4, Instituto Tecnológico de Informática, September. ESPRIT, EuTrans IT-LTR-OS-20268.

J. Andrés-Ferrer, D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. 2007. On the use of different loss functions in statistical pattern recognition applied to machine translation. To appear in Pattern Recognition Letters.

A. L. Berger, Stephen A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72, March.

P. F. Brown and other. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

P. F. Brown et al. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

Corinna Cortes, Mehryar Mohri, and Jason Weston. 2005. A general regression technique for learning transductions. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 153–160, New York, NY, USA. ACM Press.

Richard O. Duda, Peter E. Hart, and David G. Stork. 2000. *Pattern Classification*. John Wiley and Sons, New York, NY, 2nd edition.

I. García-Varea and F. Casacuberta. 2001. Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *Proc. of MT Summit VIII*, pages 115–120, Santiago de Compostela, Spain.

U. Germann et al. 2001. Fast decoding and optimal decoding for machine translation. In *Proc. of ACL01*, pages 228–235.

F. Jelinek. 1969. A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, 13:675–685.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edmonton, Canada, May.

S. Kumar and W. Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation.

J.B. Marino, R. E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà. 2006. N-gram-based machine translation. In *Computational Linguistics*, pages 527–549.

F.J. Och and H. Ney. 2004. The Alignment Template Approach to Statistical Machine Translation . *Computational Linguistics*, 30(4):417–449, December.

F. J. Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, College Park, MD, June.

F. J. Och. 2000. GIZA++: Training of statistical translation models. `http://www-i6.informatik.rwth-aachen.de/\~och/software/GIZA++.html`.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.

Kishore A. Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY, September.

V. Steinbiss R. Schlüter, T. Scharrenbach and H. Ney. 2005. Bayes risk minimization using metric loss functions. In *Proceedings of the European Conference on Speech Communication and Technology, Interspeech*, pages 1449–1452, Lisbon, Portugal, September.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133, March.

Raghavendra Udupa and Hemanta K. Maji. 2006. Computational complexity of statistical machine translation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 25–32. Trento, Italy.

N. Ueffing and H. Ney. 2004. Bayes decision rules and confidence measures for statistical machine translation. In *EsTAL - Espa for Natural Language Processing*, pages 70–81, Alicante, Spain, October. Springer Verlag, LNCS.

Ye-Yi Wang and Alex Waibel. 1997. Decoding algorithm in statistical translation. In *Proc. of ACL'97*, pages 366–372, Madrid, Spain.

A. Yaser et al. 1999. Statistical Machine Translation: Final Report. Technical report, Johns Hopkins University 1999 Summer Workshop on Language Engineering, Center for Language and Speech Processing, Baltimore, MD, USA.

R. Zens, F.J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Advances in artificial intelligence. 25. Annual German Conference on AI*, volume 2479 of *Lecture Notes in Computer Science*, pages 18–32. Springer Verlag, September.

Support Vector Machine Based Orthographic Disambiguation

Eiji ARAMAKI **Takeshi IMAI** **Kengo Miyo** **Kazuhiko Ohe**
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8655, Japan
aramaki@hcc.h.u-tokyo.ac.jp

Abstract

Orthographic variation can be a serious problem for many natural language-processing applications. Japanese in particular contains orthographic variation, because the large quantity of transliteration from other languages causes many possible spelling variations. To manage this problem, this paper proposes a support vector machine (SVM)-based classifier that can determine whether two terms are equivalent. We automatically collected both positive examples (sets of equivalent term pairs) and negative examples (sets of inequivalent term pairs). Experimental results yielded high levels of accuracy (87.8%), demonstrating the feasibility of the proposed approach.

1 Introduction

Orthographic variation can be a serious problem for many natural language-processing (NLP) applications, such as information extraction (IE), question answering (QA), and machine translation (MT). For example, many example-based machine translation (EBMT) (Nagao, 1984) methods, such as (Somers, 1999; Richardson et al., 2001; Sumita, 2001; Carl and Way, 2003; Aramaki and Kurohashi, 2004; Nakazawa et al., 2006), utilize a translation dictionary during bilingual text alignment. Also, several statistical machine translation (SMT)(Brown et al., 1993) methods set initial translation parameters using a translation dictionary. When consulting a dictionary, a system must disambiguate orthographic variation.

The following terms are an example of Japanese orthographic variation, corresponding to the term "*Avogadro's number*":

1. アヴォガドロ数
   (A **VO** GA DO RO SU),
2. アボガドロ数
   (A **BO** GA DO RO SU).

Although both terms are frequently used (term (1) resulted in 25,700 Google hits and Term (2) resulted in 25,000 Google hits[1]), translation dictionaries contain only one of the terms, resulting in low levels of accuracy with dictionary-based bilingual text alignment.

This paper focuses on Japanese orthographic disambiguation. Japanese orthographic variance is closely related to transliteration, because transliteration relies on pronunciation, the great differences between the sounds made in Japanese and in Western languages (mainly English) results in a variety of possible spellings.

Researchers have already proposed methods to solve this problem. For example, Knight(1998) developed a back-transliteration method using a probabilistic

[1]We got the results on May 14, 2007.

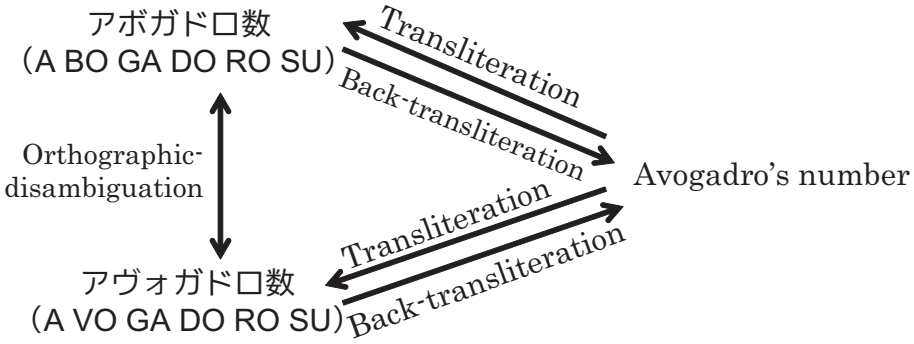


Figure 1: Transliteration and Orthographic Variation.

model. Goto et al.(2004) also developed a probabilistic model, which takes into account surrounding context. Lin and Chen(2002) developed a perceptron learning algorithm for back-transliteration. While these methods differ, they all share the same goal: being able to back-transliterate a given term into another language.

By contrast, this paper proposes a new task schema: given two Japanese terms, the system determines whether they are equivalent. Figure 1 illustrates our task schema; a foreign term can be transliterated into Japanese in several ways. While previous methods can yield suitable back-transliteration for a term, our system determines whether a pair of Japanese terms originates from the same foreign word. We expect our task-setting is more direct and practical for many applications, such as dictionary consulting in MT, IE, and so on.

For this process, our proposed method uses a machine learning technique (support vector machine, hereafter SVM (Vapnik, 1999)), which requires the two following types of data:

1. Positive examples: a term pair, which are spelled differently, but have the same meaning; and,

2. Negative examples: a term pair, which are spelled differently and have differing meanings.

While previous methods have utilized only positive examples, our proposed method also

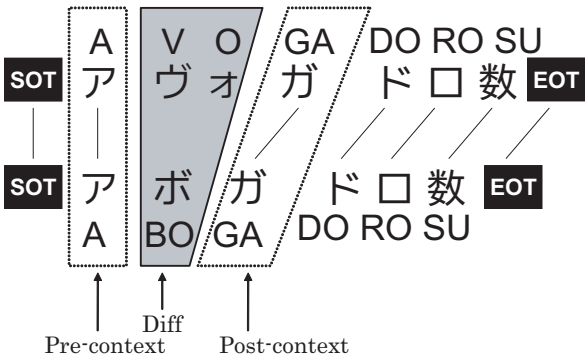


Figure 2: An Example of DIFF, PRE-CONTEXT and POST-CONTEXT.

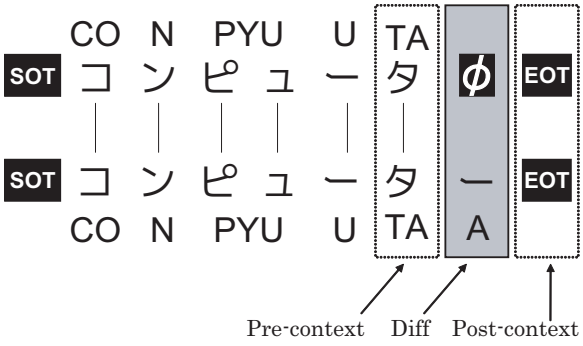


Figure 3: Another Example of DIFF, PRE-CONTEXT and POST-CONTEXT.

incorporates negative examples. Both examples can be generated automatically from translation dictionaries using spelling similarity and heuristic rules.

Experimental results yielded high accuracy (87.8%), demonstrating the feasibility of the proposed approach.

Although we investigated the performance in the medical terms, the proposed method does not depend on the target domain.

Section 2 of this paper describes how training data are built. Section 3 describes the learning method, and Section 4 presents the experimental results. Section 5 discusses related work, and Section 6 presents our conclusions.

2 Automatic Example Building

This section describes how training data are built; Section 2.1 discusses positive examples, and Section 2.2 discusses negative examples. Note that the latter is a novel task.

2.1 Positive Examples

Our method uses a standard approach to extract positive examples. The basic idea is that orthographic variants should (1) have similar spelling, and (2) share the same English translation.

The method consists of the following two steps:

STEP 1: First, using two or more translation dictionaries, we extract a set of Japanese terms with the same English translation.

STEP 2: Then, for each extracted set, we generate possible two term pairs ($term_1$ and $term_2$), and calculate the spelling similarity between them. Spelling similarity is measured using the following edit-distance based similarity $SIM(term_1, term_2)$:

$$SIM(term_1, term_2) = 1 - \frac{\text{EditDistance}(term_1, term_2) \times 2}{len(term_1) + len(term_2)},$$

where $len(term_1)$ is the length (the number of characters) of $term_1$, $len(term_2)$ is the length (the number of characters) of $term_2$, $EditDistance(term_1, term_2)$ is the minimum number of point mutations required to change $term_1$ into $term_2$, where a point mutation is one of: (1) a change in a character, (2) the insertion of a character, and (3) the deletion of a character. For details, see (Levenshtein, 1965).

Any term pair with more than a threshold (TH) similarity is considered a positive example [2].

2.2 Negative Examples

As mentioned in Section 1, generating negative examples is a novel process in this field.

One simple way is to select two words from a dictionary randomly. However, such a simple method would generate a huge quantity of meaningless examples. Therefore, as in our collection of positive examples, we collected only term pairs with similar spellings.

Another problem is a balance of the example quantity. In the preliminary experiments, the number of negative examples was about three times as the number positive examples, leading to a negative bias.

Therefore, we investigated the Google hits of each term pair by using a query, such as "アヴォガドロ数 アボガドロ数".

Then, we utilize only negative examples with many Google hits, and reject low-hits examples, because of the following two reasons:

1. **Popularity**: We expect that a more popular term pair is more informative.
2. **Reliability**: We hypothesize that an orthographic pair rarely appears in one document, because one document usually has an orthographic consistency. Therefore, we can expect that if two terms cooccur in one document, they are not orthographic variants, ensuring reliability for negative examples.

The detailed steps are as follows:

[2] We set $TH = 0.8$.

STEP 1: First, using two or more translation dictionaries, we extract a set of Japanese terms with different English translations.

STEP 2: Then, for each extracted set, we generate possible pairs, and calculate the spelling similarity between them. Any term pair exceeding a threshold (TH) similarity is considered a negative example candidate.

STEP 3: Finally, we investigate the Google hits for each candidate. We only use the top K-hits candidates as negative examples[3].

3 Leaning Method

Application of the method described in Section 2 yields training data, consisting of triple expressions $< term_1, term_2, +1/-1 >$, in which "+1" indicates a positive example (orthographic variants), and "-1" indicates a negative example (different terms). Table 1 provides some examples.

The next problem is how to convert training data into machine learning features. We regard the different parts and context (window size ± 1) as features:

1. DIFF: differing characters between two translations;
2. PRE-CONTEXT: previous character of DIFF; and
3. POST-CONTEXT: subsequent character of DIFF.

Figure 2 provides examples of these features. Since the different part is a gray area ("*VO(ヴォ)*" and "BO(ボ)"), we consider DIFF to be "VO:BO (ヴォ:ボ)" itself, PRE-CONTEXT to be "A (ア)" in a dotted box, and POST-CONTEXT to be "GA (ガ)" also in a dotted box.

Figure 3 provides another example; the insertion/deletion of a character can be considered the Diff using ϕ, such as "$\boxed{\phi}$:A ($\boxed{\phi}$:ー)". In addition, the start ($\boxed{SOT}$) or end ($\boxed{EOT}$) of a term can be considered a character.

Note that both PRE-CONTEXT and POST-CONTEXT consist of one character pair, while the DIFF can be a pair of $n : m$ characters ($n \geq 0$, $m \geq 0$).

In learning, we can use a back-off technique to prevent problems related to data sparseness. As a result, each different point utilizes the following four features:

- Diff + Pre-context + Post-context
- (1-back-off-a) Diff + Pre-context
- (1-back-off-b) Diff + Post-context
- (2-back-off) Diff

Figure 4 presents some examples.

4 Experiments

4.1 Test-set

To evaluate the performance of our system, we manually built a test-set as follows:

First, we extracted 5,013 similar spelling term pairs, that have more than ($SIM > 0.8$), from two dictionaries (Nanzando, 2001b),(Ito et al., 2003).

Then, for each pair, we annotated whether it is an equivalent pair (orthographic variants) or not (different terms).

Finally, we randomly extracted 883 pairs form it. We regard it as a test-set. The test-set consists of 312 positive examples and 571 negative examples. The others (4,130 examples) are used for training in comparative methods (BYHAND and COMBINATION mentioned in Section 4.3).

4.2 Training-set

By using the proposed method (in Section 2), we automatically built a training-set from two translation dictionaries (Japan Medical Terminology English-Japanese(Nanzando, 2001a) and 25-thousand-terms Medical Dictionary(MEID, 2005)). As a result, we got a training-set, consisting of 68,608 examples (21,380 positive examples and 47,228 negative examples).

[3] In the experiments in Section 4, we set $K = 21,380$, which is equal to the number of positive examples.

P/N*	$Term_1$	$Term_2$
+1	ヨードピラセット (YO O DO PI RA SE TTO; iodopyracet)	ヨードピラセト (YO O DO PI RA SE TO; iodopyracet)
+1	マイクロメーター (MA I KU RO ME E TA A; micrometer)	マイクロメータ (MA I KU RO ME E TA; micrometer)
+1	アンプリファイア (A N PU RI FA I A; amplifier)	アンプリファイヤー (A N PU RI FA I YA A; amplifier)
+1	オシロスコープ (O SI RO SU KO O PU; oscilloscope)	オッシロスコープ (O SSI RO SU KO O PU; oscilloscope)
+1	動コンプライアンス (DO U KO N PU RA I A N SU; dynamic compliance)	動的コンプライアンス (DO U TE KI KO N PU RA I A N SU; dynamic compliance)
+1	浸透圧性ショック (SI N TO O A TU SE I SYO KKU; osmotic shock)	浸透圧ショック (SI N TO O A TU SYO KKU; osmotic shock)
+1	マールブルグウイルス (MA A RU BU RU GU U I RU SU; Marburg virus)	マルブルグウイルス (MA RU BU RU GU U I RU SU; Marburg virus)
+1	ドールトンの法則 (DO O RU TO N NO HO O SO KU; Dalton law)	ドルトンの法則 (DO RU TO N NO HO O SO KU; Dalton law)
-1	B 型肝炎 (BI I GA TA KA N E N; hepatitis B)	C 型肝炎 (SI I GA TA KA N E N; hepatitis C)
-1	トランス (TO RA N SU; trance)	トランジスタ (TO RA N JI SU TA; transistor)
-1	ビタミン P (BI TA MI N PI I; vitamin P)	ビタミン C (BI TA MI N SI I; vitamin C)
-1	カドミウム (KA DO MI U MU; cadmium)	カルシウム (KA RU SI U MU; calcium)
-1	アルコール (A RU KO O RU; alcohol)	グルコース (GU RU KO O SU; glucose)
-1	メラトニン (ME RA TO NI N; melatonin)	セロトニン (SE RA TO NI N; serotonine)
-1	クローン (KU RO O N; clone)	クラーレ (KU RA A RE; curare)
-1	ケトン生成 (KE TO N SE I SE I; ketogenesis)	メタン生成 (ME TA N SE I SE I; methanation)
-1	リード指数 (RI I DO SI SU U; Reid index)	リビー指数 (RI BI I SI SU U; Livi index)
-1	トマチン (TO MA CHI N; tomatine)	ヘマチン (HE MA CHI N; haematin)
-1	バルーン法 (BA RU U N HO; balloon method)	ラグーン法 (RA GU U N HO; lagoon method)

Table 1: Some Examples of Training-set.

* "+1" indicates positive examples, and "-1" indicates negative examples.

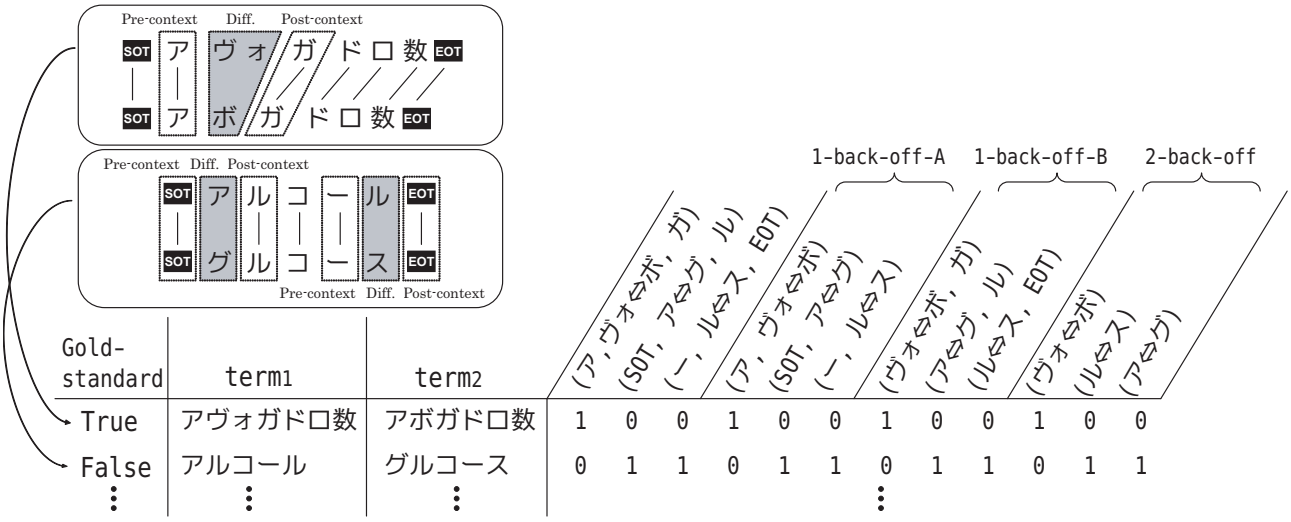


Figure 4: An Example of Features.

4.3 Comparative Methods

we compared the following methods:

1. **EDITDISTANCE(TH)**: an edit-distance-based method, which regards an example with a spelling similarity $SIM(term_1, term_2) > TH$ as an orthographic variants. The performance of this method changes, depending on TH.

2. **BYHAND**: a SVM-based method, trained by manually annotated corpus, consists of 4,130 examples.

3. **AUTOMATIC**: a SVM-based method, trained by an automatically build training-set.

4. **COMBINATION**: a SVM-based method, trained by both BYHAND corpus and AUTOMATIC corpus.

For SVM learning, we used TinySVM[4] with a linear kernel[5].

4.4 Evaluation

To evaluate our method, we used three measures, precision, recall and accuracy, defined as follows:

$$Precision = \frac{\text{\# of pairs found and correct}}{\text{total \# of pairs found}},$$

$$Recall = \frac{\text{\# of pairs found and correct}}{\text{total \# of pairs correct}},$$

$$Accuracy = \frac{\text{\# of pairs correct}}{\text{total \# of pairs in test-set}}.$$

4.5 Results

First, we checked the performance of **EDITDISTANCE(TH)** in various TH values. Figure 5 presents the results. While the precision is basically proportional to the spelling similarity (TH), it drops down in the high TH ($TH \div 0.96$), indicating a highly similar spelling term pair not always have to be the orthographic variants.

Table 2 presents the performance of all methods. AUTOMATIC did not obtain a higher accuracy than BYHAND, the combination of them is the highest accuracy, demonstrating the basic feasibility of our approach. The precision-recall graph (Figure 6) also shows the advantage of COMBINATION .

4.6 Error Analysis

We investigated the errors from COMBINATION, and found that many errors came from

[4]http://chasen.org/ taku/software/TinySVM/

[5]Although we tried a polynomial kernel and an RBF kernel, their performance are almost equal to a linear kernel.

a verbal omission, which is different phenomenon from transliteration.

For example, a test-set has the following positive example:

1. カルシウム・チャネル
   (calcium channel; KA RU SI U MU CHA NE RU),
2. カルシウム**イオン**・チャネル
   (calcium **ion** channel; KA RU SI U MU **I O N** CHA NE RU).

Because a term "*ion*" is without saying inferable in this case, it can be omitted. Capturing such an operation requires a very high level of understanding of the meaning of the terms.

To focus on a transliteration problem, we manually removed such examples from our test-set, and built a sub-set of it, consisting of only transliterations. The result is shown in Table 3. The accuracy of COMBINATION is higher than 90%.

It is difficult to compare this accuracy to that of the previous studies because (1) their corpus were different from ours and (2) previous studies focused on back-transliteration. However, we can say that the present accuracy is, at least, not behind from the previous researchers (64% by (Knight and Graehl, 1998) and 87.7% by (Goto et al., 2004)). We expect that the present accuracy is practical in many applications.

Finally, we investigate the differences between AUTOMATIC and BYHAND results (the AUTOMATIC accuracy is much lower than the BYHAND by 8.5 points in Table 2). One of the reasons is dictionary specific styles, such as numerous expression variants ("8,8, ⑧,Ⅷ,viii,ⅷ,viii,八 (Japanese number expression)"), hyphenation variants ("‐,—,=,=,･") and so on. Because the BYHAND training-set and the test-set came from the same dictionaries, BYHAND already knows such variants are meaningless differences. However, AUTOMATIC, using different dictionaries, sometimes suffered from unseen number expression/hyphenation variants.

Note that in transliteration accuracy (in Table 3), their accuracies (BYHAND and AUTOMATIC) are not so different.

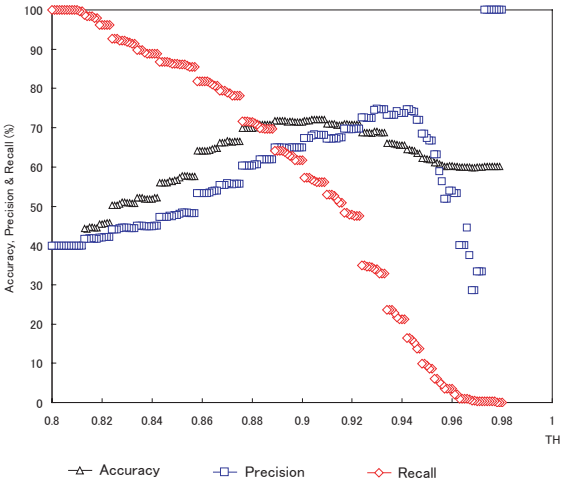


Figure 5: *TH* and EDITDISTANCE Performance.

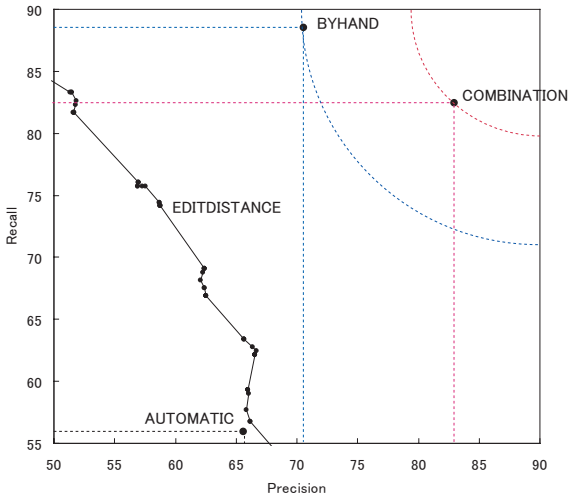


Figure 6: Precision and Recall.

Table 2: Results

methods	Precision	Recall	Accuracy
EDIT-DISTANCE(0.91)	67.2%(164/244)	52.6% (164/312)	70.9% (626/883)
BYHAND	70.4%(276/392)	**88.4% (276/312)**	82.7% (731/883)
AUTOMATIC	65.7%(177/269)	56.7% (177/312)	74.2% (656/883)
COMBINATION	**82.9%(258/311)**	82.6% (258/312)	**87.8% (776/883)**

* The performance in EDIT-DISTANCE(0.91) showed the highest accuracy in various TH values.

Table 3: Results of a sub-set (Transliteration Only)

methods	Precision	Recall	Accuracy
BYHAND	67.7%(122/180)	**91.0%(122/134)**	80.3% (286/356)
AUTOMATIC	77.3%(109/141)	81.3% (109/134)	83.9% (299/356)
COMBINATION	**90.6%(117/129)**	90.7% (117/134)	**91.9% (327/356)**

5 Related Works

As noted in Section 1, transliteration is the field most relevant to our work, because many orthographic variations come from borrowed words. Our proposed method differs from previous studies in the following three ways: (1) task setting, (2) negative examples, and (3) target scope.

5.1 Task Setting

Most previous studies have involved finding the most suitable back-transliteration of a term.

For example, given an observed Japanese string o by optical character recognition (OCR) software, Knight and Graehl (1998) finds a suitable English word w. For this process, they developed a probabilistic model that decomposed a transliteration into sub-operations as follows:

$$P(w)P(e|w)P(j|e)P(k|j)P(o|k),$$

where $P(w)$ generates written English word sequences, $P(e|w)$ pronounces English word sequences, $P(j|e)$ converts English sounds into Japanese sounds, $P(k|j)$ converts Japanese sounds to KATAKANA writing, and $P(o|k)$ introduces misspellings caused by OCR.

While this method is phoneme-based, Bilac and Tanaka(2004) combined phoneme-based and graphme-based transliteration. Goto et al.(2004) proposed a similar method, utilizing the surrounding context.

Such methods are not only applicable to Japanese; it can also be used for Arabic(Stalls and Knight, 1998; Sherif and Kondrak, 2007), Chinese(Li et al., 2007), Persian(Karimi et al., 2007).

The task-setting involved in our method differs from previous methods. Our methodology involves determining whether two terms in the same language are equivalent, making our task-setting more direct and suitable than previous methods for many applications, such as dictionary consulting in MT and information retrieval.

Note that Yoon et al.(2007) also proposed a discriminative transliteration method, but their system determines whether a target term is transliterated from a source term or not.

5.2 Negative Examples

Our task setting requires negative examples, consisting of term pairs with similar spellings, but different meanings.

By contrast, previous research involved only positive examples. For example, Masuyama et al.(2004) collected 178,569 Japanese transliteration variants (positive examples) from large corpora. However, they paid little attention to negative examples.

5.3 Target Scope

As mentioned above, orthographic variation in Japanese results mainly from transliteration. However, our target includes several different phenomena, such as verbal omissions mentioned in Section 4.6. Although the accuracy for omissions is not enough, our method addresses it easily, while previous methods are unable to handle this kind of phenomenon.

6 Conclusion

In this paper, we proposed a SVM-based orthographic disambiguation method. We also proposed a method for collecting both positive and negative examples. Experimental results yielded high levels of accuracy (87.8%), demonstrating the feasibility of the proposed approach.

Acknowledgments

Part of this research is supported by Grant-in-Aid for Scientific Research of Japan Society for the Promotion of Science (Project Number:16200039, F.Y.2004-2007 and 18700133, F.Y.2006-2007) and the Research Collaboration Project (#047100001247) with Japan Anatomy Laboratory Co.Ltd.

References

Eiji Aramaki and Sadao Kurohashi. 2004. Example-based machine translation using structual translation examples. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT2004)*, pages 91–94.

Slaven Bilac and Hozumi Tanaka. 2004. A hybrid back-transliteration system for Japanese. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 597–603.

Peter F. Brown, Stephen A. Della Pietra, Vi cent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Michael Carl and Andy Way. 2003. *Recent Advances in Example-based Machine Translation*. Kluwer Academic Publishers.

Isao Goto, Naoto Kato, Terumasa Ehara, and Hideki Tanaka. 2004. Back transliteration from Japanese to English using target English context. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 827–833.

M. Ito, H. Imura, and H. Takahisa. 2003. *IGAKU-SHOIN'S MEDICAL DICTIONARY*. Igakusyoin.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2007. Collapsed consonant and vowel models: New approaches for English-Persian transliteration and back-transliteration. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 648–655.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

V. I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

Haizhou Li, Khe Chai Sim, Jin-Shea Kuo, and Minghui Dong. 2007. Semantic transliteration of personal names. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 120–127.

Wei-Hao Lin and Hsin-Hsi Chen. 2002. Backward machine transliteration by learning phonetic similarity. In *Proceeding of the 6th conference on Natural language learning*, pages 1–7.

Takeshi Masuyama, Satoshi Sekine, and Hiroshi Nakagawa. 2004. Automatic construction of Japanese KATAKANA variant list from large corpus. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 1214–1219.

MEID. 2005. *25-Mango Medical Dictionary*. Nichigai Associates, Inc.

Makoto Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In *In Artificial and Human Intelligence*, pages 173–180.

Toshiaki Nakazawa, Kun Yu, Daisuke Kawahara, and Sadao Kurohashi. 2006. Example-based machine translation based on deeper NLP. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT2006)*, pages 64–70.

Nanzando. 2001a. *Japan Medical Terminology English-Japanese 2nd Edition*. Committee of Medical Terminology, NANZANDO Co.,Ltd.

Nanzando. 2001b. *Promedica ver.3*. NANZANDO Co.,Ltd.

Stephen D. Richardson, William B. Dolan, Arul Menezes, and Monica Corston-Oliver. 2001. Overcoming the customization bottleneck using example-based MT. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL2001) Workshop on Data-Driven Methods in Machine Translation*, pages 9–16.

Tarek Sherif and Grzegorz Kondrak. 2007. Substring-based transliteration. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 944–951.

Harold Somers. 1999. Example-based machine translation. In *Machine Translation*, pages 113–157.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating names and technical terms in arabic text. In *Proceedings of The 17th International Conference on Computational Linguistics (COLING1998) Workshop on Computational Approaches to Semitic Languages*.

Eiichiro Sumita. 2001. Example-based machine translation using dp-matching between word sequences. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL2001) Workshop on Data-Driven Methods in Machine Translation*, pages 1–8.

Vladimir Vapnik. 1999. *The Nature of Statistical Learning Theory*. Springer-Verlag.

Su-Youn Yoon, Kyoung-Young Kim, and Richard Sproat. 2007. Multilingual transliteration using feature based phonetic method. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 112–119.

Breaking the barrier of context-freeness. Towards a linguistically adequate probabilistic dependency model of parallel texts.

Matthias Buch-Kromann
ISV Computational Linguistics Group
Copenhagen Business School
DK-2000 Frederiksberg, Denmark
mbk.isv@cbs.dk

Abstract

This paper presents a generative probabilistic dependency model of parallel texts that can be used for statistical machine translation and parallel parsing. Unlike syntactic models that are based on context-free dependency grammars, the dependency model proposed in this paper is based on a sophisticated notion of dependency grammar that is capable of modelling non-projective word order and island constraints, the complement-adjunct distinction, as well as deletions and additions in translations.

1 Introduction

Dependency grammar has attracted much attention in computational linguistics in recent years. In statistical machine translation, several researchers have proposed SMT systems that are based on dependency grammars, including (Fox, 2005; Quirk et al., 2005; Ding, 2006; Smith and Eisner, 2006; Hall and Němec, 2007). However, the dependency-based SMT systems that have been proposed in the literature are almost uniformly based on projective (usually context-free) dependency grammars, ie, grammars that disallow the kind of crossing dependencies shown in Figure 1 and explained in section 3.

From a linguistic point of view, the projectivity assumption is unfortunate because non-projectivity is a high-frequent phenomenon that manifests itself in long-distance phenomena such as topicalization, scrambling, and extraposition.

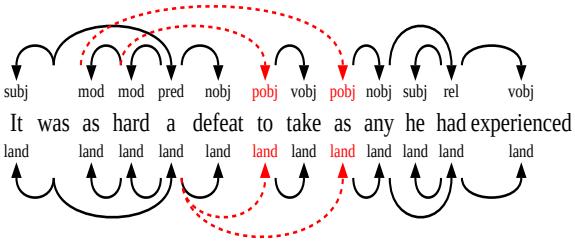


Figure 1: Authentic example with a doubly non-projective dependency tree and corresponding surface structure. Dependency and landing edges for non-projective nodes are shown with dashes.

Eg, in the dependency treebanks for Slovene, Arabic, Dutch, Czech, and Danish, 0.4–5.4% of all dependencies are non-projective, and 11.2–36.4% of all sentences contain a non-projective dependency (Nilsson et al., 2007). Since it is difficult to model non-projective word orders correctly with projective syntax models, and such errors often result in meaning-disturbing translation errors, non-projectivity is more important than its relatively small contribution to precision and recall in monolingual parsing suggests. (Buch-Kromann, 2006, sections 1.4, 2.4, 4.2) gives a more comprehensive list of linguistic constructions that are difficult to model within a projective setting.

Within a monolingual setting, there are many dependency frameworks that account for most of these phenomena, including Word Grammar (Hudson, 2007), Functional-Generative Description (Sgall et al., 1986), Weighted Constraint Dependency Grammar (Schröder, 2002), Extensible Dependency Grammar (Debusmann et al., 2004), and Discontinuous Grammar (Buch-Kromann, 2006). But, as far as we know, none of these de-

pendency frameworks have so far provided a linguistically well-motivated non-projective dependency framework for parallel texts, and done so within a probabilistic setting. This is a gap that we hope to fill with the present paper.

The paper is structured as follows. In section 2, we describe how machine translation and parallel parsing can be viewed as optimization problems within a generative probabilistic dependency model of parallel texts. In section 3, we describe our notion of parallel dependency analyses and how they are used to control word order. In section 4, we introduce our notion of translation units. In section 5, we describe our generative probabilistic dependency model of parallel texts. In section 6, we briefly outline some ideas for how grammar induction can be carried out within our framework. Section 7 presents our conclusions.

2 Statistical dependency-based translation and parallel parsing

From an abstract point of view, a *parallel probabilistic dependency grammar* can be viewed as a probability measure $P(\mathcal{A})$ on the space $\mathbb{A}$ of all conceivable parallel dependency analyses. In this setting, machine translation and parallel parsing can be reduced to the problem of optimizing $P(\mathcal{A})$ with different side conditions.

In *translation*, we know a source text t and need to find the most probable parallel dependency analysis, $\mathrm{Trans}(t)$, that matches t. That is, we must find:

$$\mathrm{Trans}(t) = \arg\max_{\substack{\mathcal{A}\in\mathbb{A} \\ Y(\mathcal{A})=t}} P(\mathcal{A})$$

where $Y(\mathcal{A})$ denotes the source text associated with $\mathcal{A}$, and $Y'(\mathcal{A})$ the target text. Once we have computed $\mathrm{Trans}(t)$, it is easy to compute the optimal translation by extracting the target text from $\mathrm{Trans}(t)$ by means of Y'.[1]

Similarly, in *parallel (synchronous) parsing* — which is essential for turning a parallel corpus into a parallel dependency treebank — we know a source text t and a target text t', and need to find the most probable parallel dependency analysis, $\mathrm{Parse}(t, t')$, that matches the given source and target texts t, t'. That is, we must find:

$$\mathrm{Parse}(t, t') = \arg\max_{\substack{\mathcal{A}\in\mathbb{A} \\ Y(\mathcal{A})=t \\ Y'(\mathcal{A})=t'}} P(\mathcal{A}).$$

In our generative probability model, we assume that a parallel dependency analysis $\mathcal{A}$ consists of a source text analysis D, a target text analysis D', and a word alignment W. We will factor:

$$P(\mathcal{A}) = P(D, D', W) = P(D) \cdot P(D', W|D)$$

and model the monolingual source analysis probability $P(D)$ and the translation probability $P(D', W|D)$ separately. Note that unlike the probability model in phrase-based SMT (Koehn et al., 2003), where the source text is generated from the target text, our probability model follows the natural direction of translation. This is also the approach used in the probability model by (Smith and Eisner, 2006), but for projective rather than non-projective dependency grammars.

The asymmetry between source and target language in our model is sensible from a linguistic point of view, since it is well-known among translation scholars that translations tend to differ significantly from normal texts in the target language. This asymmetry means that our translation model resembles a transfer-based system in important respects. However, unlike traditional transfer systems, the model does not require the parallel parser or translation system to make a hard choice about the source language analysis before deciding on a target language analysis.

Several problems must be solved in order to build a functioning parallel parser or machine translation system that uses these ideas to circumvent the linguistic limitations of projective dependency grammars: we must (a) formulate a linguistically sensible notion of parallel dependency analyses and parallel probabilistic dependency grammars; (b) specify a method for inducing such grammars from parallel corpora and/or parallel dependency treebanks; and (c) identify computationally efficient optimization algorithms

[1] In the SMT literature, the translation t' of t is often defined as the target text t' that maximizes $P(t'|t) = \sum_{\mathcal{A}\in\mathbb{A}\text{ s.t. }Y(\mathcal{A})=t, Y'(\mathcal{A})=t'} P(\mathcal{A}|t)$. From a linguistic point of view, there is no solid argument for preferring one definition over the other, and by looking for the optimal parallel analysis rather than the optimal target text, we avoid the computationally difficult problem of calculating the sum.

for translation and parallel parsing that normally succeed in finding optimal or near-optimal translations and parallel parses. This paper focuses on (a), and largely ignores (b) and (c). More information about our solution to (b) and (c) is presented in (Buch-Kromann, 2007a; Buch-Kromann, 2007b). Our analyses are based on the dependency framework Discontinuous Grammar (Buch-Kromann, 2006).

3 Parallel dependency analyses

In a parallel dependency analysis $\mathcal{A} = (D, D', W)$, each word alignment $w \leftrightarrow w'$ in W is assumed to encode a translational correspondence between the word clusters w and w' in the source text and target text, ie, the word alignment encodes the intuition that the subset w of words in the source text corresponds roughly in meaning or function to the subset w' of words in the target text. The translations may contain additions or deletions, ie, w and w' may be empty.

The monolingual dependency analyses D and D' are assumed to consist of dependency edges linking the words in the text. Each dependency edge $d \xleftarrow{r} g$ encodes a complement or adjunct relation between a word g (the *governor*) and a complement or adjunct phrase headed by the word d (the *dependent*), where the edge label r specifies the complement or adjunct dependency role.[2] In our analyses, the dependencies in the source analysis are required to form a tree (or a forest), and similarly with the dependencies in the target analysis. Moreover, our parallel dependency analyses must be well-formed with respect to translation units, in a sense that is described briefly in section 4 and defined formally in (Buch-Kromann, 2007a).

Figure 2 shows an example of this kind of analysis, based on the annotation conventions used in the Copenhagen Danish-English Dependency Treebank (Buch-Kromann, 2007a). In the example, word alignments are indicated by lines connecting Danish word clusters with English word

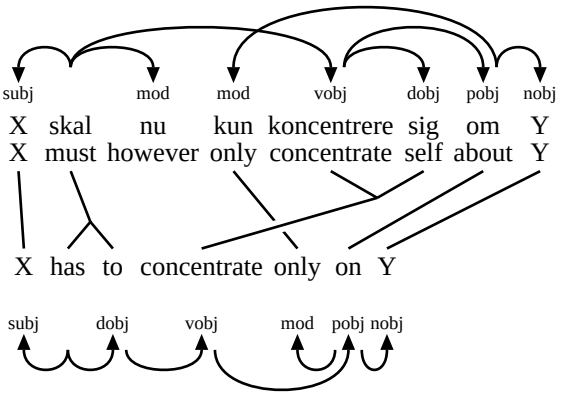


Figure 2: Parallel dependency treebank analysis with word alignment and two monolingual dependency analyses (with non-projective word order).

clusters, and dependencies are indicated by means of arrows that point from the governor to the dependent, with the dependency role written at the arrow tip. For example, the Danish word cluster "koncentrere sig" ("concentrate self") has been aligned with the English word "concentrate", and the English phrase headed by "on" is analyzed as a prepositional object of the verb "concentrate."[3]

In order to model word order and island constraints, each word w in the source and target dependency trees is assigned a *landing site* l, defined as the lowest transitive governor of w that dominates all words between w and l; a node w that has l as its landing site is called a *landed node* of l, and the landing relation between w and l is encoded by means of a *landing edge* $w \xleftarrow{\text{land}} l$. If the governor g and landing site l of a word w do not coincide ($g \neq l$), then the dependency edge $w \leftarrow g$ is called *non-projective*; otherwise, it is called *projective*. In projective dependency grammars, we always have $g = l$. Figure 1 shows an example of a dependency tree with two non-projective dependency edges ('to $\xleftarrow{\text{pobj}}$ hard' and 'as $\xleftarrow{\text{pobj}}$ as'). The word "a" functions as the landing site for both "hard" and "as" because it is the lowest transitive governor that dominates all the nodes between these two words and their respec-

[2] Following standard dependency theoretic assumptions, we assume: (a) complements are lexically licensed by their governor, whereas adjuncts license their adjunct governor; (b) in the functor-argument structure, complements act as arguments of their governor, whereas adjuncts act as modifiers; (c) a governor can have several adjuncts with the same adjunct role, whereas complement roles must be unique.

[3] Dependency analyses differ from phrase-structure analyses in that phrases are a derived notion: in a dependency tree, each word has a derived phrase that consists of all the words that can be reached from the word by following the arrows. For example, the English word "concentrate" heads the phrase "concentrate only on Y," and the Danish word "om" heads the discontinuous phrase "kun . . . om Y."

tive governors.

It can be shown that the landing edges associated with a dependency tree always form a projective tree, called the *surface tree*. The projectivity allows landing sites to control the global word order by controlling the local relative word order of their landed nodes — ie, landing sites have the word ordering responsibility assigned to governors in projective dependency grammars.

The *extraction path* for a word w is defined as the shortest path from the governor g to the landing site l of w. For example, in Figure 1, the word "to" has extraction path 'hard $\xleftarrow{\text{mod}}$ a', and the second "as" ("as$_2$") has extraction path 'as$_1$ $\xleftarrow{\text{mod}}$ hard $\xleftarrow{\text{mod}}$ a'. As argued by (Buch-Kromann, 2006, p. 98), extraction paths are useful for modelling island constraints in a dependency-based setting. For example, the adjunct island constraint states that nothing may be moved out of an adverbial adjunct, which corresponds to the claim that an extraction path cannot contain an adjunct edge of the form $x \leftarrow y$ where y is a verb.

4 Syntactic translation units[4]

In order to define our notion of syntactic translation units, we need to introduce the following terminology. The definitions below apply to both source and target words and dependencies. Two words are said to be *coaligned* if they belong to the same alignment edge. A dependency edge $d \xleftarrow{r} g$ is called *internal* if d and g are coaligned, and *external* otherwise. A word w is called *singular* if it fails to be coaligned with at least one word in the other language. By an abuse of terminology, we will say that a word d is a *dependent* of an alignment edge $w \leftrightarrow w'$ provided d is a dependent of some word in $w \cup w'$ and d is not itself contained in $w \cup w'$. For example, in Figure 2, the words "has", "to", and "skal" are coaligned, the dependency 'to $\xleftarrow{\text{dobj}}$ has' is internal, the dependency 'concentrate $\xleftarrow{\text{vobj}}$ to' is external, the word "nu" is singular, and the word "X" is a dependent of the alignment edge "skal $\leftrightarrow$ has to".

The *translation unit* corresponding to the word alignment $w \leftrightarrow w'$ is defined as the subgraph of the analysis $\mathcal{A}$ consisting of all nodes in $w \cup w'$, all internal dependency and alignment edges within $w \leftrightarrow w'$, and all external dependencies of $w \leftrightarrow w'$ except for parallel and singular adjuncts, where the external dependents are replaced with argument variables $x_1, \ldots, x_n$ and $x'_1, \ldots, x'_{n'}$. Figure 3 shows the six translation units that can be derived from the parallel dependency analysis in Figure 2 in this way. Each translation unit can be interpreted as a bidirectional translation rule: eg, the first translation unit in Figure 3 can be interpreted as a translation rule stating that a Danish dependency tree with terminals "x_1 skal x_2" can be translated into an English dependency tree with terminals "x'_1 has to x'_2" where the English phrases x'_1, x'_2 are translations of the Danish phrases x_1, x_2, and vice versa.

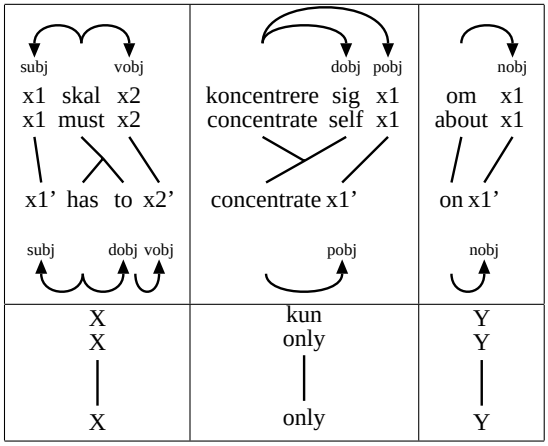


Figure 3: The six translation units derived from the parallel dependency analysis in Figure 2.

In order to have a meaningful interpretation as a translation rule, a translation unit must have a parallel set of source and target argument variables, and a well-formed source and target dependency analysis, as defined formally in (Buch-Kromann, 2007a). In general, parallel dependency treebanks are not guaranteed to lead to translation units that satisfy these requirements. However, (Buch-Kromann, 2007a) has defined an algorithm that can compute a *minimal reduction* that is computed by merging word alignments in a minimal way, in which the resulting translation units satisfy the requirements. As an example of how this procedure works, Figure 4 shows a head-switching example (left) borrowed from (Way, 2001), and the corresponding minimal reduction (right) computed by the merging algo-

[4]This section is based on (Buch-Kromann, 2007a).

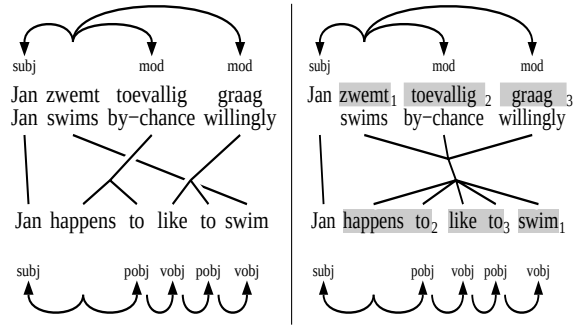


Figure 4: A head-switching example (left) and the associated minimal reduction (right).

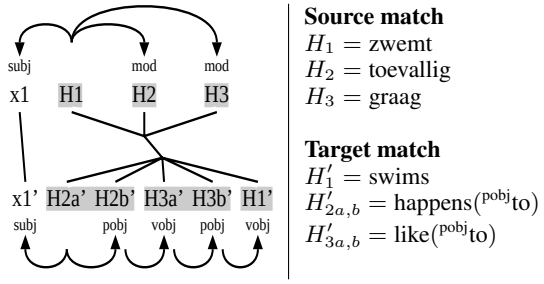


Figure 5: Syntactic translation template induced from Figure 4, with source and target match.

procedure probabilistic graph generation
begin
recursively expand source root TOP (cf. Figure 7)
recursively translate source root TOP (cf. Figure 8)
return *generated graph and probability*
end

Figure 6: Our probabilistic graph generation procedure (a Markov process).

Top-down expansion of source node w_i
S1. Identify landing site and relative word order
S2. Select complement frame
S3. Generate and recursively expand complements
S4. Generate and recursively expand adjuncts

Figure 7: The steps in the top-down expansion of a source word w_i in our generative probabilistic dependency model.

rithm, with the original word alignments indicated by means of the numbered boxes. We can think of the original word alignments in the treebank as *lexical translation units* (the smallest lexically meaningful units of translation), and of the merged word alignments as *syntactic translation units* (the smallest syntactically meaningful units of translation).

In this paper, we will for simplicity assume that each syntactic translation unit consists of a single lexical translation unit. However, a more elegant and general account of head-switching phenomena can be provided by decomposing syntactic translation units by means of their original lexical translation units. Eg, instead of using zwemt($^{\text{mod}}$toevallig,$^{\text{mod}}$graag) $\leftrightarrow$ happens($^{\text{pobj}}$to($^{\text{vobj}}$like($^{\text{pobj}}$to($^{\text{vobj}}$swim)))) as an atomic lexical translation unit in the translation of the example in Figure 4, we can decompose the translation into several steps by first matching the source analysis with the abstract syntactic translation template shown in Figure 5, and then deciding on the choice of lexical translation units in a target language top-down manner: ie, we first select "toevallig $\leftrightarrow$ happen($^{\text{pobj}}$to)" as a translation of "toevallig", then "graag $\leftrightarrow$ like($^{\text{pobj}}$to)", and finally "zwemt $\leftrightarrow$ swim."

5 A generative probabilistic dependency model of parallel texts

We will now present a generative probabilistic dependency model of parallel texts that models complements, adjuncts, landing sites, local word order, island constraints, and additions and deletions during translation. The source dependency model is a simplification of (Buch-Kromann, 2006, ch. 6) in that we ignore secondary dependencies, gapping coordinations, antecedents, and punctuation. We assume that the source and target analyses have formal root nodes TOP and TOP′ (aligned with each other), and that all words in the source and target text are transitive dependents of the top nodes; in particular, the root of a sentence in the source and target analysis is assumed to be a *root* adjunct of the top node.

The generative procedure is modelled as a top-down Markov process (Figure 6). The generative procedure first creates the source tree by recursively expanding TOP in steps S1–S4, and then creates the target tree and the word alignments by recursively translating TOP in steps T1–T5. The individual steps in the source and target node expansion are shown in Figures 7 and 8, and described in detail below. In our dependency model, the probability of a parallel dependency analysis

Top-down translation of source node w_i
T1. Identify landing sites and word order in target tunit
T2. Generate and recursively expand tunit arguments
T3. Identify deleted source adjuncts
T4. Generate and recursively translate parallel adjuncts
T5. Generate added target adjuncts

Figure 8: The steps in the top-down translation of a source word w_i in our generative probabilistic dependency model.

Notation	Meaning
w_i	ith word (source > 0, target < 0)
d_i	dependency role of ith word
$cframe_i$	complement frame at w_i
$aframe_i$	adjunct roles at w_i
g_i	governor of w_i
l_i	landing site of w_i
o_i	relative word order of w_i at l_i
path(w_i, w_j)	upwards path from node w_i to transitive governor w_j
τ_i	syntactic translation unit for w_i
S_i	source analysis for τ_i
T_i	target analysis for τ_i
$w_{i'}$	target root of τ_i
int_i	internal source nodes in τ_i
int'_i	internal target nodes in τ_i
$extadj_i$	external adjuncts of τ_i
$added_i$	added external target adjuncts of τ_i
$args_i$	source arguments of τ_i

Figure 9: The notation used to refer to the governor, landing site, word order, etc. of a source or target node N.

$\mathcal{A}$ is computed by

$$P(\mathcal{A}) = P_{\text{S}}(\text{TOP}) \cdot P_{\text{T}}(\text{TOP})$$

where P_{S} and P_{T} are defined recursively by $P_{\text{S}}(w_i) = P_{\text{S1}}(w_i) \cdots P_{\text{S4}}(w_i)$ and $P_{\text{T}}(w_i) = P_{\text{T1}}(w_i) \cdots P_{\text{T5}}(w_i)$, using the probabilities for steps S1–S4 and T1–T5 defined below.

In the following, given a source or target node w_i (with source nodes having $i > 0$, target nodes $i < 0$), we will use the notation shown in Figure 9. By an abuse of notation, we will use w_i^* to denote the set of all relevant covariates associated with w_i when w_i is expanded or translated; the covariates may include any aspects of the structure that have been generated at the given point in the generation, including (but not necessarily restricted to) all relevant node features and dependency roles of w_i, l_i, g_i, etc. Determining the right set of covariates for each of the distributions in our model is an empirical question which we will ignore in the rest of this paper.

5.1 Modelling source analyses

The steps S1–S4 are used to model node expansion in source analyses. Steps S2–S4 are similar in spirit to the steps proposed by (Eisner, 1996; Collins, 1997) for statistical dependency parsing, whereas the submodel S1 for island constraints and local word order is new.

S1. Identify landing site and word order

The first step in the source expansion of w_i is to choose a landing site l_i among the transitive governors of w_i, and a linear ordering o_i that indicates the word order of w_i at l_i relative to the previously landed nodes at l_i.[5] For each possible landing site l and word order o, we want to quantify how well-formed that choice of landing site and word order is with respect to (a) island constraints expressed in terms of the extraction path from g_i to l, and (b) the local word order position o assigned to w_i at l.

As noted in section 3, an extraction can be blocked by the presence of island edges in the extraction path (eg, adjunct edges with verbal governors). Island edges can be detected statistically by observing that if an edge $x \xleftarrow{r} y$ occurs less often in extraction paths than in the treebank in general, then the edge is likely to be an island edge, ie, the blocking effect of an edge $x \xleftarrow{r} y$ for the word w_i can be modelled by means of:

$$\min\left(1, \frac{P_{\text{extpath}}(x \xleftarrow{r} y | w_i)}{P_{\text{deptree}}(x \xleftarrow{r} y)}\right)$$

where the minimum ensures that non-island edges cannot improve the global extraction probability. P_{extpath} is the probability distribution of edges in extraction paths, and P_{deptree} is the probability distribution of edges in dependency trees. Ie, the relative probability $E_{w_i,l}$ of the extraction path produced by choosing l as the landing site of w_i is expressed by:

$$E_{w_i,l} = \prod_{\substack{(x \xleftarrow{r} y) \in \\ \text{path}(g_i,l)}} \min\left(1, \frac{P_{\text{extpath}}(x \xleftarrow{r} y | w_i)}{P_{\text{deptree}}(x \xleftarrow{r} y)}\right)$$

[5] Following (Buch-Kromann, 2006, pp. 276-277), we assume that dependencies are generated in a predefined derivation order. Nodes that precede the current landed node in the derivation order are called *previously landed nodes*.

In order to model the probability of the local word order position, we note that the choice of local word order o for w_i at l can be modelled as a process where w_i is inserted at position o, and the dummy node STOP is inserted at all other positions so that we can detect the absence of an obligatorily present node. If we let $P_{\text{worder}}(w|c_{l,o})$ denote the probability of inserting word or dummy word w as a landed node at a position o with word order context $c_{l,o}$, then the relative probability $O_{w_i,l,o}$ of the choice of local word order o for w_i at l is expressed by:

$$O_{w_i,l,o} = P_{\text{worder}}(w|c_{l,o}) \prod_{o' \neq o} P_{\text{worder}}(\text{STOP}|c_{l,o'})$$

(Buch-Kromann, 2006, section 6.2) has proposed a local word order context that only includes the neighbouring complements, the neighbouring adjuncts, the landing site, and a binary variable that indicates whether the position is to the left or right of the landing site. These covariates suffice to encode a wide range of local word order constraints, such as "adverbials cannot be inserted between a verb and an adjacent subject," "a verb does not allow two simultaneous complements on its left," and "a finite verb requires a subject to its left," but in probabilistic rather than absolute terms.

With the relative probability of the extraction path quantified by $E_{w_i,l}$ and the relative probability of the local word order quantified by $O_{w_i,l,o}$, we can compute the probability of the actual choice of l_i, o_i by normalizing the probabilities, ie by setting:

$$P_{\text{S1}}(w_i) = \frac{E_{w_i,l_i} \cdot O_{w_i,l_i,o_i}}{\sum_{l,o} E_{w_i,l} \cdot O_{w_i,l,o}}.$$

As argued by (Buch-Kromann, 2006, section 7.3), under linguistically reasonable assumptions about island constraints and the number of complements and adjuncts that a word can have, a landing site has a bounded number of landing positions, and a word has at most $\log n$ landing sites where n is the number of words in the graph. The sum can therefore be computed efficiently in $O(\log n)$ time.

S2. Select complement frame

In step 2 of the source expansion, we must choose a complement frame *cframe*$_i$ for w_i. This choice can be modelled by means of

$$P_{\text{S2}}(w_i) = P_{\text{cframe}}(\textit{cframe}_i|w_i^*)$$

where $P_{\text{cframe}}(\textit{cframe}|w_i^*)$ is the probability of generating the complement frame *cframe* at w_i.

S3. Generate and expand complements

In step 3 of the source expansion, we must choose a complement word w_j for each complement role d_j in *cframe*$_i$, and expand the complement recursively. We model this by:

$$P_{\text{S3}}(w_i) = \prod_{d_j \in \textit{cframe}_i} P_{\text{comp}}(w_j|d_j, w_i^*) P_{\text{S}}(w_j)$$

where $P_{\text{comp}}(w|d, w_i^*)$ is the probability of generating the complement w for complement role d at w_i.[6]

S4. Generate and expand adjuncts

In step 4 of the source expansion, we must generate the adjuncts of w_i and expand them recursively. We model this as a process where the governor generates a list of adjunct roles *aframe*$_i$ at w_i one by one with probability $P_{\text{arole}}(d_j|w_i^*)$, until the special adjunct role STOP is generated with probability $P_{\text{arole}}(\text{STOP}|w_i^*)$. As each adjunct role d_j is generated, we generate an adjunct word w_j with probability $P_{\text{adj}}(w_j|d_j, w_i^*)$ and expand w_j recursively, ie, the adjuncts of w_i are generated with probability:

$$P_{\text{S4}}(w_i) = P_{\text{arole}}(\text{STOP}|w_i^*) \cdot \prod_{d_j \in \textit{aframe}_i} P_{\text{arole}}(d_j|w_i^*) P_{\text{adj}}(w_j|d_j, w_i^*) P_{\text{S}}(w_j)$$

5.2 Modelling the translation from source analyses to target analyses

The steps T1–T5 are used to model the translation from source analyses to target analyses. Probability distributions for the target language are indicated by means of primes. Eg, $P_{\text{S1}'}(w_i)$ denotes the probability of the monolingual expansion step S1 at the target word w_i, but for the target language rather than the source language.

[6] Although we could have designed a model that can learn statistical dependencies between different complement slots, we use a simpler model where the complements are generated independently of each other. The simple model is justified by (Li and Abe, 1999), who report that the statistical dependencies between complement roles are rather weak, and therefore difficult to detect.

T1. Identify landing site and relative word order in target unit

In step T1, we must identify landing sites and relative word order for the internal target nodes int'_i in the syntactic translation unit τ_i with source root w_i. If the target word order is assumed to be completely independent of the source word order, we can simply define:

$$P_{\text{T1}}(w_i) = \prod_{w_j \in int'_i} P_{\text{S1}'}(w_j)$$

where int'_i is processed in the target language derivation order.

However, languages tend to place discourse-old material in the beginning of sentences, and discourse-new material in the end. It therefore often makes sense to use the source word order as a guide to target word order. This can be accomplished by including the relative ordering of the source nodes corresponding to the target nodes within the target word order context $c'_{l,o}$.

T2. Generate and translate tunit arguments

In step T2, we need to recursively translate the source arguments $args_i$ of the translation unit τ_i. For each $w_j \in args_i$ we select a translation unit τ_j that matches the source analysis at w_j. Like in noisy-channel SMT, we must balance the adequacy A and fluency F of our choice of τ_j at w_j, ie, we must try to find a compromise between the admissibility of τ_j as a translation of the source tree in τ_j (*adequacy*) and the admissibility of the target tree in τ_j as a target subtree at the target root $w_{j'}$ of τ_j (*fluency*).

We can model the adequacy of τ_j as a translation of the source tree at w_j by means of the probability:

$$A(w_j, \tau_j) = P_{\text{tunit}}(\tau_j | w_j^*)$$

where $P_{\text{tunit}}(\tau | w_j^*)$ is the probability of translating a source structure at w_j by means of the matching translation unit τ.

Similarly, we can model the fluency of the source tree T_j at the target root $w_{j'}$ by means of the probability:

$$F(w_j, \tau_j) = P_{\text{comp}'/\text{adj}'}(w_{j'} | d_{j'}, w_{j'}^*) \cdot P_{\text{S}'_{234}}(T_j)$$

where $P_{\text{comp}'/\text{adj}'}$ denotes either $P_{\text{comp}'}$ or $P_{\text{adj}'}$, depending on whether $w_{j'}$ is a complement or an adjunct, and where $P_{\text{S234}'}(T_j)$ denotes the monolingual target language probability of the target dependency tree T_j without any word order (ie, steps S2′–S4′ only).

Like in noisy-channel SMT, we can compromise between adequacy and fluency by weighing them by means of the formula $A^\lambda F^{1-\lambda}$ for some $\lambda \in [0, 1]$. Setting λ close to 1 results in translations with high adequacy and low fluency, and vice versa when setting λ close to 0. We can therefore model the probability of choosing the translation unit τ_j to transfer the source tree at w_j by means of:

$$P_{\text{transfer}}(w_j, \tau_j) = \frac{A(w_j, \tau_j)^\lambda F(w_j, \tau_j)^{1-\lambda}}{\sum_\tau A(w_j, \tau)^\lambda F(w_j, \tau)^{1-\lambda}}.$$

This allows us to model:

$$P_{\text{T2}}(w_i) = \prod_{w_j \in args_i} P_{\text{transfer}}(w_j, \tau_j) P_{\text{T}}(w_j).$$

T3. Identify deleted source adjuncts

In step T3, we need to decide for each external source adjunct w_j in $extadj_i$ whether w_j should be deleted in the translation ($\delta_j = 1$) or translated into the target language ($\delta_j = 0$). In general, it is not a good idea to delete content words in the translation. However, there are sometimes mismatches in the translation, and there are also some aspects of syntax, especially discourse particles and punctuation, that are language-specific and consequently often ignored in the translation. We will therefore include deletions in our model, by defining:

$$P_{\text{T3}}(w_i) = \prod_{w_j \in extadj_i} P_{\text{del}}(\delta_j | w_j^*)$$

where $P_{\text{del}}(\delta_j = 1 | w_j^*)$ is the probability of deleting the adjunct w_j in the translation.

T4. Generate and translate parallel adjuncts

For each non-deleted external source adjunct w_j in $extadj_i$ (ie, each w_j where $\delta_j = 0$), we need to (a) select a target adjunct role $d_{j'}$ and a target adjunct governor $g_{j'}$ within the target tree T_i, (b) select a translation unit τ_j that matches the source analysis at w_j, and (c) expand w_j recursively.

In step T4a, we want to quantify the probability of the chosen target adjunct governor $g_{j'}$ and role $d_{j'}$, given the corresponding source adjunct governor g_j and role d_j. The relative probability of a particular choice (g', d') can be modelled statistically by assigning large weight to choices of (g', d') that occur above chance level, and low weight to choices that occur below chance level, ie, the relative probability of the choice (g', d') can be expressed by the quantity

$$I_{g',d'|g,d} = \frac{P_{\text{adjtrans}}(g', d'|g, d)}{P_{\text{adjtrans}}(g', d')}$$

where $P_{\text{adjtrans}}(g', d')$ is the probability that a parallel adjunct has target governor g' and target role d', and $P_{\text{adjtrans}}(g', d'|g, d)$ is the same probability with the conditional knowledge that the parallel adjunct has source governor g and source role d. By normalizing the weights, we can compute:

$$P_{\text{T4a}}(w_j) = \frac{I_{g_{j'},d_{j'}|g_j,d_j}}{\sum_{g',d'} I_{g',d'|g_j,d_j}}.$$

In step T4b, we must select a translation unit τ_j for each non-deleted adjunct w_j, given the target adjunct role $d_{j'}$ and target adjunct governor $g_{j'}$. This is modelled exactly as in step T2, but for non-deleted external source adjuncts rather than translation unit arguments.

Combining (a) and (b), we therefore define:

$$P_{\text{T4}}(w_i) = \prod_{\substack{w_j \in \mathit{extadj}_i \\ \delta_j = 0}} P_{\text{T4a}}(w_j) P_{\text{T4b}}(w_j) P_{\text{T}}(w_j).$$

T5. Generate added adjuncts

In step T5, we must generate the added target adjuncts in the target analysis. We do this by traversing the internal target nodes in int'_i in target derivation order: for each internal target node w_j in int'_i, we (a) generate a sequence added_j of added target adjunct phrases one at a time, until the special stop symbol STOP is generated, and (b) assign landing sites to the generated target adjunct phrases in the process.

Step T5a can therefore be computed by:

$$P_{\text{T5a}}(w_j) = P_{\text{add-arole}}(\text{STOP}|w_j^*) \cdot \prod_{w_k \in \mathit{added}_j} P_{\text{add-arole}}(d_k|w_j^*) P_{\text{add-adj}}(T_k|d_k, w_j^*)$$

where $P_{\text{add-arole}}(d|w_j^*)$ is the probability of creating an added target adjunct with adjunct role d at w_j, and $P_{\text{add-adj}}(T|d_k, w_j^*)$ is the probability of creating the added target adjunct tree T given adjunct role d_k at w_j. T5b can be computed by means of:

$$P_{\text{T5b}}(w_j) = \prod_{w_k \in \mathit{added}_j} P_{\text{T1'}}(T_k)$$

where $P_{\text{T1'}}(T_k)$ is the probability of the target landing sites assigned to the words in the target adjunct phrase T_k.

We therefore have:

$$P_{\text{T5}}(w_i) = \prod_{w_j \in \mathit{int}'_i} P_{\text{T5a}}(w_j) P_{\text{T5b}}(w_j).$$

6 Statistical estimation and optimization

Our generative probabilistic dependency model decomposes the probability of the entire analysis into probabilities associated with individual steps in the generative procedure, such as P_{cframe}, P_{extpath}, $P_{\text{add-adj}}$, etc. Each of these distributions can be estimated from parallel dependency treebank data by means of any suitable density estimator, including Generalized Linear Models and Generalized Additive Models (which have log-linear models as a special case) and the XHPM estimator proposed by (Buch-Kromann, 2006, ch. 5,6). The XHPM estimator is a generalization of (Li and Abe, 1999) that is designed specifically for categorical data equipped with classification hierarchies. As a correction estimator, the XHPM estimator may be particularly suited to estimating probability ratios of the form $P(x|y)/P(x)$, which is needed in steps S1 and T4.

7 Conclusions

In this paper, we have presented a generative probabilistic dependency model of parallel texts that can be used for machine translation and parallel parsing. Unlike previous dependency models used in machine translation, the proposed model is not based on context-free dependency grammar, but builds on a more sophisticated notion of dependency theory that is capable of modelling complements and adjuncts, non-projective dependencies and island constraints, as well as deletions and additions in the translation. In this respect, our model can be seen as a step towards

translation models that are more realistic from a linguistic point of view. By allowing syntactic translation units to be arbitrarily large parallel tree structures, and decomposing syntactic translation units into lexical translation units, the model may even provide an elegant account of head-switching.

There are many issues that need to be addressed before the dependency model we have presented can be used to build a functioning machine translation or parallel parsing system. First of all, we have not described how to estimate the many probabilities in our dependency model from parallel treebank data. Secondly, some empirical work remains to be done with respect to choosing the relevant covariates in each generative step. Finally, although (Buch-Kromann, 2007b) has started work in this direction, we still need to develop a computationally efficient algorithm that is capable of computing optimal or near-optimal solutions to the optimization problems posed by parallel parsing and machine translation.

8 Acknowledgements

This work was supported by a grant from the Danish Research Council for the Humanities. Thanks to the anonymous reviewers for their careful reviews and highly valuable advice.

References

Matthias Buch-Kromann. 2006. Discontinuous Grammar. A dependency-based model of human parsing and language learning. Dr.ling.merc. dissertation, Copenhagen Business School. http://www.id.cbs.dk/∼mbk/thesis.

Matthias Buch-Kromann. 2007a. Computing translation units and quantifying parallelism in parallel dependency treebanks. In *Proc. of Linguistic Annotation Workshop, ACL-2007*. See errata on http://www.isv.cbs.dk/∼mbk/pub/2007-law.html.

Matthias Buch-Kromann. 2007b. Dependency-based machine translation and parallel parsing without the projectivity and edge-factoring assumptions. a white paper. Working paper, ISV Computational Linguistics Group, Copenhagen Business School. URL www.isv.cbs.dk/ mbk/pub/2007-white.pdf.

Michael Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proc. ACL-1997*, pages 16–23.

Ralph Debusmann, Denys Duchier, and Geert-Jan Kruijff. 2004. Extensible Dependency Grammar: A new methodology. In *Proc. Recent Advances in Dependency Grammar, COLING-2004*.

Yuan Ding. 2006. *Machine translation using Probabilistic Synchronous Dependency Insertion Grammars*. Ph.D. thesis, Univ. of Pennsylvania.

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proc. COLING-96*, pages 340–345.

Heidi J. Fox. 2005. Dependency-based statistical machine translation. In *Proc. 2005 ACL Student Workshop*.

Keith Hall and Petr Němec. 2007. Generation in machine translation from deep syntactic trees. In *Proc. CoNLL-2007*.

Richard Hudson. 2007. *Language Networks. The new Word Grammar*. Oxford University Press.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. HLT-NAACL-2003*.

Hang Li and Naoki Abe. 1999. Learning dependencies between case frame slots. *Computational Linguistics*, 25(2):283–291.

Jens Nilsson, Joakim Nivre, and Johan Hall. 2007. Generalizing tree transformations for inductive dependency parsing. In *Proc. ACL-2007*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proc. ACL-2005*.

Ingo Schröder. 2002. *Natural language parsing with graded constraints*. Ph.D. thesis, Univ. of Hamburg.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. Reidel, Dordrecht.

David A. Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proc. HLT-NAACL Workshop on Statistical Machine Translation*, pages 23–30.

Andy Way. 2001. Solving headswitching translation cases in LFG-DOT. In *Proc. LFG-2001*.

Demonstration of the German to English METIS-II MT System

Michael Carl, Sandrine Garnier and Paul Schmidt
Institut für Angewandte Informationsforschung,
66111 Saarbrücken, Germany
{carl,sandrine,paul}@iai.uni-sb.de

Starting in October 2004, METIS-II is the continuation of the successful project METIS I (IST-2001-32775). Like METIS I, METIS II aims at translation of free text input based on rule-based, statistical and pattern matching methods. The METIS-II project has four partners, translating from their 'home' languages Greek, Dutch German, and Spanish into English.

The following goals and premises were defined for the project:

1. use of a bilingual hand-made dictionary
2. use of 'basic' NLP tools and resources
3. different tag-sets for SL and TL possible
4. translation units below the sentence border
5. use a monolingual target language corpus
6. no bilingual corpus required

In particular, the availability of the monolingual target language corpus makes METIS-II a data-driven MT system. However, parallel corpora as in SMT/EBMT are not required. For our German-to-English METIS-II system we have designed and implemented an architecture which uses rule-based devices to generate sets of partial translation hypotheses and a statistical *Ranker* to evaluate and retrieve the best hypotheses in their context[1]. Similar architectures have already been suggested as EBMT systems (Sato and Nagao, 1990), for instance with their MBT2 system. Methods to integrate knowledge bases and statistics have also been explored in (Knight et al., 1994) and recently in the LOGON-project (Oepen et al., 2007) which uses statistical feature functions to select the best rule-induced structures at various stages during processing.

In the German-to-English METIS-II system, rule-based devices generate an acyclic AND/OR graph which allows for compact representation of many different translations while the *Ranker* is a beam search algorithm which tries to find most likely paths through the AND/OR graph. The architecture consists of the following five steps:

1. The *Analyser* lemmatises and morphologically analyses the SL sentence. It produces a (flat) grammatical analysis of the sentence, detecting phrases and clauses and potential subject candidates. The *Analyser* uses the linguistic technology available at the IAI.
2. *Dictionary Lookup* matches analysed SL sentence on the transfer dictionary and retrieves TL equivalences. This procedure retrieves ambiguous and/or overlapping entries and stores them in a partial OR graph. Our German to English dictionary contains more than 629.000 single and multi-word entries. Since matching proceeds on morphemes and lemmatised forms, a sophisticated compilation of the dictionary into a database is required. As described in (Carl and Rascu, 2006), the matching procedure is also suited to retrieve discontinuous entries.
3. The *Expander* inserts, deletes, moves and permutes items or chunks in the graph generated by the *Dictionary Lookup* according to TL syntax. The *Expander* is a rule-based device and extends the AND/OR graph with further partial translation hypotheses. It is called *Expander* because it expands the search space with additional paths. The operations of the *Expander* and its modifications on the graph are such that each path through the graph consumes exactly once the translation(s) of each word of the source language sentence. For our German-to-English implementation we have currently ca. 50 rules.
4. The *Ranker* is a beam search algorithm that iteratively traverses the AND/OR graph and computes the most likely translations in a log-linear fashion (Och and Ney, 2002). Unlike a usual statistical decoder (Koehn, 2004) — but similar to the method suggested by (Knight et al., 1994) — our *Ranker* traverses the search graph to grade alternative paths and outputs a list of the n-best translations. The *Ranker* itself does not modify the graph. It does not permute chunks or items and it does not generate additional paths which are not already contained in

[1] A full description of the system is provided in (Carl, 2007).

the graph.

5. A *Token Generator* generates surface wordforms from the lemmas and PoS tags. The Token Generator has been described in (Carl et al., 2005).

The *Ranker* and the *Token Generator* are trained on the British National Corpus (BNC[2]). It is a collection of tagged texts making use of the CLAWS5 tag set which comprises roughly 70 different tags[3]. The heuristic functions of the *Ranker* are trained with the CMU-language modelling toolkit.

Evaluation In a first experiment we have tested the system on four languages (Dutch, German, Greek and Spanish) into English based on 50 sentences for each of the languages. The results are shown in table (1). A separate set of *Expander* rules was developed for each source language, consisting of five rules for Greek up to approx. 20 rules for German.

Language	BLEU	NIST
Dutch	0.4034	6.4489
Spanish	0.3701	5.7304
Greek	0.2138	5.1220
German	0.1671	3.9197

Table 1: Results of first Experiment

Another set of evaluations was conducted one a German test set of 200 sentences after enhacing the *Dictionary Lookup*, *Expander*, and *Ranker* modules. Our best results are shown in the first line in table (2). However, they (still) lag behind those produced by Systran (Babelfish) on the same test set, as shown last line in table (2).

NIST	BLEU	token model	tag model
5.3193	0.2231	5M-n3	5M-n7
6.3644	0.3133	—	—

Table 2: Results of 200 test translations

A full description of the system is provided in (Carl, 2007).

References

Michael Carl and Ecaterina Rascu. 2006. A dictionary lookup strategy for translating discontinuous phrases. In *Proceedings of the 11th EAMT Conference*, pages 49–58, Oslo, Norway.

Michael Carl, Paul Schmidt, and Jörg Schütz. 2005. Reversible Template-based Shake & Bake Generation. In *In Proceedings of the Example-Based Machine Translation Workshop held in conjunction with the 10th Machine Translation Summit*, pages 17–26, Phuket, Tailand.

Michael Carl. 2007. METIS-II: The German to English MT System. In *Proceedings of the 11th Machine Translation Summit*, Copenhagen, Denmark.

Kevin Knight, Ishwar Chandler, Mattew Haines, Vasileios Hatzivassiloglou, Edouard Hovy, Masayo Ida, K. Luk, Steve, Akitoshi Okumura, Richard Whitney, and Kenji Yamada. 1994. Integrating Knowledge Bases and Statistics in MT. In *Proceedings of the AMTA*.

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*, pages 115–124.

Franz J. Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th annual ACL Conference*, pages 295–302, Philadelphia, PA.

Stephan Oepen, Erik Velldal, Jan Tore Lonning, Paul Meurer, and Victoria Rosen. 2007. Towards Hybrid Quality-Oriented Machine Translation – On Linguistics and Probabilities in MT. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Skövde, Sweden.

Satoshi Sato and Makato Nagao. 1990. Towards memory-based translation. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*, pages 247–252, Helsinki, Finnland.

[2]The BNC consists of more than 100 million words in more than 6 million sentences `http://www.natcorp.ox.ac.uk/`

[3]To reach reversibility of the lemmatiser, and discriminate between otherwise ambiguous analyses, we have added a few tags.

How Phrase Sense Disambiguation outperforms Word Sense Disambiguation for Statistical Machine Translation

Marine CARPUAT marine@cs.ust.hk
Dekai WU dekai@cs.ust.hk

Human Language Technology Center
HKUST
Department of Computer Science and Engineering
University of Science and Technology, Clear Water Bay, Hong Kong

Abstract

We present comparative empirical evidence arguing that a generalized *phrase sense disambiguation* approach better improves statistical machine translation than ordinary word sense disambiguation, along with a data analysis suggesting the reasons for this. Standalone word sense disambiguation, as exemplified by the Senseval series of evaluations, typically defines the target of disambiguation as a single word. But in order to be useful in statistical machine translation, our studies indicate that word sense disambiguation should be redefined to move beyond the particular case of single word targets, and instead to generalize to multi-word phrase targets. We investigate how and why the phrase sense disambiguation approach—in contrast to recent efforts to apply traditional word sense disambiguation to SMT—is able to yield statistically significant yimprovements in translation quality even under large data conditions, and consistently improve SMT across both IWSLT and NIST Chinese-English text translation tasks. We discuss architectural issues raised by this change of perspective, and consider the new model architecture necessitated by the phrase sense disambiguation approach.

*This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under GALE Contract No. HR0011-06-C-0023, and by the Hong Kong Research Grants Council (RGC) research grants RGC6083/99E, RGC6256/00E, and DAG03/04.EG09. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency.

1 Introduction

Until recently, attempts to apply word sense disambiguation (WSD) techniques to improve translation quality in statistical machine translation (SMT) models have met with mixed or disappointing results (e.g., Carpuat and Wu (2005), Cabezas and Resnik (2005)), suggesting that a deeper empirical exploration of the differences and consequences of the assumptions of WSD and SMT is called for.

On one hand, word sense disambiguation as a standalone task consists in identifying the correct sense of a given word among a set of predefined sense candidates. In the Senseval series of evaluations, WSD targets are typically single words, both in the lexical sample tasks, where only a predefined set of targets are considered (e.g., Kilgarriff (2001);), and in the all-words tasks, where all content word in a given corpus must be disambiguated (e.g., Kilgarriff and Rosenzweig (1999)).

This focus on single words as WSD targets might be explained by the sense inventory, which is usually derived from a manually constructed dictionary or ontology, where most entries are single words. In addition, historically, as for many other tasks, work on European languages imposed whitespace as an easy way to define convenient

word boundaries. Linguistically, however, this oversimplistic modeling approach seems rather questionable, and recalls long-held debates over the issue of what properly constitutes a "word".

In contrast, work in statistical machine translation has for some time recognized the need to segment sentences as required by the task's evaluation criteria, and today most systems use phrases or segments, and not single words, as the basic unit for lexical choice (e.g., Wu (1997); Och and Ney (2004); Koehn (2004); Chiang (2005)). Note that single-word based SMT architectures already perform a significant amount of sense disambiguation intrinsically, by virtue of combining a priori sense candidate likelihoods (from adequacy criteria as modeled by lexical translation probabilities) with contextual coherence preferences (from fluency criteria as modeled by language model probabilities). Phrasal SMT architectures, furthermore, integrate lexical collocation preferences into the disambiguation choices, raising the bar yet higher.

This suggests that to be effective at improving disambiguation accuracy within SMT architectures, sense disambiguation techniques may need to incorporate assumptions at least as strong as those already made by the SMT models. Dedicated WSD models do appear to possess traits that are promising for SMT: they employ a much broader range of features for sense selection than SMT models, and are far more sensitive to dynamic context. The question, however, is whether these advantages must be reformulated within a phrasal framework in order for the advantages to be realizable for SMT.

In this work, we empirically compare the efficacy of *phrase sense disambiguation* versus word sense disambiguation approaches toward improving translation quality of SMT models. The phrase sense disambiguation (PSD) approach generalizes word sense disambiguation to multi-word targets, aiming thereby to incorporate the crucial assumptions responsible for the success of phrasal SMT approaches into the sense disambiguation model as well. Our results and analysis show that it is indeed necessary to move away from the simplistic single-word level definition of sense disambiguation targets, in order to be useful to SMT. In effect, this argues for redefining WSD for the task of SMT. This task-driven approach to sense disambiguation requires several changes:

- Sense disambiguation targets are very different from Senseval targets.
- Sense candidates are not extracted from manually defined sense inventories, but from automatically annotated data.
- Sense disambiguation predictions require a dynamic integration architecture in SMT systems in order to be useful.

We will begin by reviewing our phrase sense disambiguation approach for SMT and contrasting it against previous word-based models. We then describe new contrastive empirical studies aimed at directly assessing the differences. On one hand, we note that incorporating multi-word PSD into phrasal SMT reliably and consistently improves translation quality, as measured by *all* eight most commonly used evaluation metrics, on *all* four different test sets from the IWSLT and NIST Chinese-English translation tasks. On the other hand, the contrastive experiments reported here show that incorporating single-word WSD into phrasal SMT leads to unpredictable and inconsistent effects on translation quality, depending on which evaluation metric one looks at. We then turn to data analysis exploring more closely how and why the multi-word PSD approach outperforms the single-word WSD approach. The analysis shows that dynamic integration of PSD prediction is crucial to this improvement, as it allows all PSD predictions to participate in the segmention of the input sentence that yields the best translation quality.

2 Previous work

In Carpuat and Wu (2007), we proposed a novel general framework for integrating a generalized sense disambiguation method into SMT, such that phrasal lexical choice is dynamically influenced by context-dependent probabilities or scores. This Phrase Sense Disambiguation—as opposed to Word Sense Disambiguation—approach appears to be the only model to date that has been shown capable of consistently yielding improvements on translation quality across all

different test sets and automatic evaluation metrics. Other related work has all been heavily oriented toward disambiguating single words.

In perhaps the earliest study of WSD potential for SMT performance by Brown *et al.* (1991), the authors reported improved translation quality on a French to English task, by choosing an English translation for a French word based on the single contextual feature which is reliably discriminative. However, this was a pilot study, which is limited to *single words* with exactly two translation candidates, and it is far from clear that the conclusions could generalize to more recent SMT architectures. In contrast with Brown *et al.*'s work, our approach incorporates the predictions of state-of-the-art WSD models (generalized to PSD models) that use rich contextual features for *any* phrase in the input vocabulary.

More recent work on WSD systems designed for the specific purpose of translation has followed the traditional word-based definition of the WSD task. Vickrey *et al.* (2005) train a logistic regression WSD model on data extracted from automatically word aligned parallel corpora, and evaluate it on a blank filling task, which is essentially an evaluation of WSD accuracy. Specia *et al.* (2007) use an inductive logic programming based WSD system to integrate expressive features for Portuguese to English translation, but this system was also only evaluated on WSD accuracy, and not integrated in a full-scale machine translation system. Even when using automatically-aligned SMT parallel corpora to define WSD tasks, as in the SemEval-2007 English Lexical Sample Task via English-Chinese Parallel Text (Ng and Chan, 2007), WSD is still defined as a word-based task.

There have been other attempts at using context information for lexical selection in SMT, but the focus was also on single words vs. multi-word phrases, and they were not evaluated in terms of translation quality. For instance, Garcia-Varea *et al.* (2001) and Garcia-Varea *et al.* (2002) show improved alignment error rate with a maximum entropy based context-dependent lexical choice model, but do not report improved translation accuracy. Another problem in the context-sensitive SMT models of Garcia Varea *et al.* is that they strictly reside within the Bayesian source-channel model, which is word-based.

The few recent attempts at integrating *single word* based WSD models into SMT have failed to obtain clear improvements in terms of translation quality. Carpuat and Wu (2005) show that using word-based Senseval trained models does not help BLEU score when integrated in a standard word-based translation system, for a NIST Chinese-English translation task.

Following this surprising result, a few attempts at integrating WSD methods into state-of-the-art SMT systems have begun to obtain slightly more encouraging results by moving away from manually-constructed sense inventories, and instead automatically defining word senses as word translation candidates, just like in SMT. Cabezas and Resnik (2005) reported that incorporating *word-based* WSD predictions via the Pharaoh XML markup scheme yielded a small improvement in BLEU score over a *phrasal* SMT baseline on a single Spanish-English translation data set. However, the result was not statistically significant, and in this paper, we will show that applying a similar single-word based model to several Chinese-English datasets does not yield systematic improvements on most MT evaluation metrics. Carpuat *et al.* (2006) also reported small improvements in BLEU score by using single-word WSD predictions in a Pharaoh baseline. However, these small improvements were obtained on a slightly weaker SMT baseline, and subsequent evaluations showed that these gains are not consistent across metrics. Giménez and Màrquez (2007) also used WSD predictions in Pharaoh for the slightly more general case of very frequent phrases, which in practice essentially limits the set of WSD targets to single words or very short phrases. However, evaluation on the single Europarl Spanish-English task did not yield consistent improvements across metrics: BLEU score did not improve, while there were small improvements in the QUEEN, METEOR and ROUGE metrics. Chan *et al.* (2007) report an improved BLEU score for a hierarchical phrase-based SMT system on a NIST Chinese-English task, by incorporating WSD predictions only for single words and short phrases of length 1 or 2. However, no results for metrics other than BLEU were reported, and no results on other tasks, so the relia-

bility of this model is not known.

What the foregoing attempts at WSD in SMT share is that (1) they focus on single words rather than full phrases, and (2) the evaluations do not show consistent improvement systematically across different tasks and metrics.

In contrast, we showed in Carpuat and Wu (2007) for the first time that generalizing WSD to exactly match *phrasal* lexical choice in SMT yields consistent improvements on 4 different test sets as measured by 8 common automatic evaluation metrics, unlike all the single-word oriented approaches. The key question left unanswered, however—which we attempt to address in the present paper—is exactly how and why it is necessary to generalize Word Sense Disambiguation to Phrase Sense Disambiguation in order to obtain this sort of consistency in translation accuracy improvement.

3 Building multi-word Phrase Sense Disambiguation models for SMT

3.1 Phrase sense disambiguation vs. word sense disambiguation

In a task-driven definition of sense disambiguation for phrase-based SMT, the PSD approach argues that disambiguation targets must be exactly the same *phrases* as in the SMT phrasal translation lexicon, so that the sense disambiguation task is identical to lexical choice for SMT. This constrasts with the standalone WSD perspective, where targets are single words, as in Senseval tasks (e.g., Kilgarriff and Rosenzweig (1999)). In SMT, phrases are typically defined as any sequence of words up to a given length. As a result, the phrasal targets for sense disambiguation need not necessarily be syntactic well-formed phrases, but rather need only be collocations defined by their surface form. This again departs from Senseval-style WSD where POS-tagging is typically decoupled from WSD, as training data is manually checked to contain instance for a single POS of the target.

In sense disambiguation for SMT, the sense candidates are those defined by the SMT translation lexicon. Sense candidates can be single words or multi-word phrases regardless of the length of the target. Note that phrasal senses do occasionally also exist in standalone WSD tasks. For instance, the Senseval English Lexical Sample tasks include WordNet phrasal senses (e.g., "polar bear" is a sense candidate for the English target word "bear".)

Given the above definitions for sense disambiguation targets and senses, annotated training data can naturally be drawn from the automatically aligned parallel corpora used to learn the SMT lexicon. Given a Chinese-English sentence pair, a WSD or PSD target in the Chinese sentence is annotated with the English phrase which is consistent with the word alignment. The definition of consistency with the word alignment should be exactly the one used for building the SMT lexicon.

Despite the differences introduced by the use of phrasal targets, the disambiguation task remains in the character and spirit of WSD. The translation lexical choice problem is exactly the same task as in recent and coming Senseval Multilingual Lexical Sample tasks (e.g., Chklovski *et al.* (2004)), where sense inventories represent the semantic distinctions made by another language. In our SMT-driven approach to PSD rather than WSD, we are only generalizing the definition of the sense disambiguation targets, and automating the sense annotation process.

3.2 Leveraging Senseval classifiers for both WSD and PSD

As in Carpuat and Wu (2007), the word sense disambiguation system is modeled after the best performing WSD system in the Chinese lexical sample task at Senseval-3 (Carpuat *et al.*, 2004). The features employed include position-sensitive, syntactic, and local collocational features, and are therefore much richer than those used in most SMT systems.

4 Integrating multi-word PSD vs. single-word WSD into phrasal SMT architectures

Unlike single-word WSD, it is non-trivial to incorporate the PSD predictions into an existing phrase-based architecture such as Pharaoh (Koehn, 2004), since the decoder is not set up to easily accept multiple translation probabilities that are dynamically computed in context-

sensitive fashion. While PSD and WSD models differ in principle only by the length of the WSD target, their integration into phrase-based SMT architectures requires significantly different strategies.

Since multi-word PSD predictions are defined for every entry in the SMT lexicon or phrase table, they can be thought of as an additional feature in the phrase table. However, unlike baseline SMT translation probabilities, these predictions are context-sensitive, and require to be updated for every new sentence. Therefore, instead of using a static phrasal translation lexicon, integration of PSD predictions require dynamically updating the phrasal translation lexicon for each sentence during decoding.

In contrast, in the single-word WSD system, since the WSD predictions only cover a subset of the phrase-table entries and the word-based targets do not have overlapping spans, it is usually possible to implement a much simpler integration architecture, by annotating the input sentence to contain the WSD predictions, as with the Pharaoh XML markup scheme.

Thus, the dynamic phrase table architecture for PSD integration necessarily generates a significant overhead. While we could in theory annotate the input sentence with phrase-based WSD predictions, just like for single-word based WSD, we argue that this approach is not optimal and would in fact hurt translation quality: annotation schemes such as the Pharaoh XML markup do not allow to annotate overlapping spans, and would thus require to commit to a phrasal segmentation of the input sentence *before* decoding. It is impossible to find an optimal phrasal segmentation before decoding, since the quality of the segmentation can only be evaluated by the translation it yields.

5 Comparative experiment setup

5.1 Data set

In order to better isolate the different effects of WSD versus PSD, comparative experiments are conducted using training and evaluation data drawn from the multilingual BTEC corpus, which contains sentences used in conversations in the travel domain, and their translations in several languages. The simpler character of these sentences faciliates clearer identification of individual factors in data analysis, compared with open domain newsire text where too many factors interfere with each other. We used a subset of this data which was made available for the IWSLT 2006 evaluation campaign; the training set consists of 40000 sentence pairs, and each test set contains around 500 sentences. We used only the pure text data, so that speech-specific issues would not interfere with our primary goal of understanding the effect of integrating WSD/PSD in a full-scale phrasal SMT model.

We also report results of the large scale evaluation of the PSD model conducted on the standard NIST Chinese-English test set (MT-04), which contains 1788 sentences drawn from newswire corpora, and is therefore of a much wider domain than the IWSLT data set.

5.2 Baseline SMT system

Since our focus is not on a specific SMT architecture, we use the off-the-shelf phrase-based decoder Pharaoh (Koehn, 2004) trained in a standard fashion on the IWSLT training set, as in Carpuat and Wu (2007).

5.3 WSD and PSD models

WSD classifiers are trained for every word, while PSD classifiers are trained for every multi-word phrase in the test set vocabularies. The number of targets is therefore much higher than even in the all-words WSD tasks. For the first IWSLT test set which contains 506 sentences, we have a total of PSD 2882 targets, as opposed to only 948 WSD targets. There is on average 7.3 sense candidates and 79 training instances per PSD target.

The scale of WSD and PSD models for SMT greatly constrasts with, for instance, the Senseval-3 Chinese lexical sample task which considered only 21 single word targets, with an average of 3.95 senses and 37 training instances per target.

6 Comparative evaluation results

The comparative experiments clearly show a marked difference between single–word WSD and multi-word PSD results. Evaluation scores, summarized in Tables 2 and 3, show that multi-word PSD yields consistent improvements in

Table 1: Evaluation results on the IWSLT-07 dataset: integrating the WSD translation predictions for single words has unpredictable effects on BLEU, NIST, METEOR, WER, PER, CDER and TER across all 3 different available test sets. Using only more reliable target words, such as nouns and verbs only, or targets that have more than 30 training instances, does not yield clear improvement either.

Test Set	Experiment	BLEU	NIST	METEOR	METEOR (no syn)	TER	WER	PER	CDER
#1	Baseline	42.21	7.888	65.40	63.24	40.45	45.58	37.80	40.09
	+WSD (all words)	41.94	7.911	65.55	63.52	40.59	45.61	37.75	40.09
	+WSD (nouns and verbs)	42.19	7.920	65.97	63.88	40.64	45.88	37.58	40.14
	+WSD (>30)	42.08	7.902	65.43	63.30	40.52	45.57	37.80	40.06
#2	Baseline	41.49	8.167	66.25	63.85	40.95	46.42	37.52	40.35
	+WSD (all words)	41.31	8.161	66.23	63.72	41.34	46.82	37.98	40.69
	+WSD (nouns and verbs)	41.25	8.135	66.08	63.40	41.30	46.76	37.85	40.65
	+WSD (>30)	41.56	8.186	66.44	63.89	40.87	46.36	37.57	40.35
#3	Baseline	49.91	9.016	73.36	70.70	35.60	40.60	32.30	35.46
	+WSD (all words)	49.73	9.017	73.32	70.82	35.72	40.61	32.10	35.30
	+WSD (nouns and verbs)	49.58	9.003	73.07	70.46	35.94	40.84	32.40	35.62
	+WSD (>30)	50.11	9.043	73.60	70.98	35.41	40.38	32.23	35.30

Table 2: Evaluation results on the IWSLT-06 dataset: integrating the multi-word PSD translation predictions for all phrases improves BLEU, NIST, METEOR, WER, PER, CDER and TER across all 3 different available test sets. In contrast, using the traditional single-word WSD approach has an unreliable impact on translation quality.

Test Set	Experiment	BLEU	NIST	METEOR	METEOR (no syn)	TER	WER	PER	CDER
#1	Baseline	42.21	7.888	65.40	63.24	40.45	45.58	37.80	40.09
	+WSD (all words)	41.94	7.911	65.55	63.52	40.59	45.61	37.75	40.09
	+PSD (all phrases)	**42.38**	**7.902**	**65.73**	**63.64**	**39.98**	**45.30**	**37.60**	**39.91**
#2	Baseline	41.49	8.167	66.25	63.85	40.95	46.42	37.52	40.35
	+WSD (all words)	41.31	8.161	66.23	63.72	41.34	46.82	37.98	40.69
	+PSD (all phrases)	**41.97**	**8.244**	**66.35**	**63.86**	**40.63**	**46.14**	**37.25**	**40.10**
#3	Baseline	49.91	9.016	73.36	70.70	35.60	40.60	32.30	35.46
	+WSD (all words)	49.73	9.017	73.32	70.82	35.72	40.61	32.10	35.30
	+PSD (all phrases)	**51.05**	**9.142**	**74.13**	**71.44**	**34.68**	**39.75**	**31.71**	**34.58**

translation quality, across *all* metrics and on *all* test sets, including statistically significant improvements on the large NIST task, while in contrast, the impact of single-word WSD on translation quality is highly unpredictable. In particular, the single-word WSD results are inconsistent across different test sets, and depend on which evaluation metric is chosen.

In order to measure the impact of WSD on translation quality, the translation results were evaluated using *all eight* of the most commonly used automatic evaluation metrics. In addition to the widely used BLEU (Papineni *et al.*, 2002) and NIST (Doddington, 2002) scores, we also evaluate translation quality with METEOR (Banerjee and Lavie, 2005), Word Error Rate (WER), Position-independent word Error Rate (PER) (Tillmann *et al.*, 1997), CDER (Leusch

et al., 2006), and Translation Edit Rate (TER) (Snover *et al.*, 2006). Note that we report METEOR scores computed both with and without using WordNet synonyms to match translation candidates and references, showing that the improvement is not due to context-independent synonym matches at evaluation time.

In the sections that follow, we investigate various reasons that PSD outperforms WSD, drawing from data analysis on these comparative experiments.

7 Single-word WSD yields unreliable results

Using WSD predictions for all the single words in a given test set has an unreliable impact on translation quality, as can be seen in Table 1. While it yields a very small, non-significant gain on NIST and METEOR on Test Set 1, it yields worse BLEU, NIST and METEOR scores for all the other test sets.

In order to check that this disappointing result cannot be simply explained by the effect of unusual target words, we perform two sets of additional experiments. We attempt to consider only target words that are closer to those used in Senseval evaluations for which these WSD models were initially designed, and demonstrated good performance.

Instead of using WSD predictions for all the whitespace separated tokens that were seen during training, we restrict our set of WSD targets to nouns and verbs. This is slightly closer to the definition of targets in Senseval tasks, which typically include nouns, verbs and sometimes adjectives, but never punctuation or any function word. Table 1 shows that this does not help translation quality compared to the baseline system, and actually underperforms using WSD predictions for all words.

In contrast with Senseval target words, which are picked so that representative training data can be obtained, we are using every target word in the vocabulary, whatever the available training data. In order to check that the target words with few training instances are not hurting the contribution of other targets, we try to restrict our set of target words to those for which at least 30 instances were seen during training. Table 1 shows that this does not have a reliable effect on translation quality either, yielding small gains in BLEU, NIST and METEOR scores over the baseline for Test Sets 2 and 3, but hurting BLEU on Test Set 1. While the results are overall slightly better than when using all WSD predictions for all words, there is no clear trend for improvement.

These results show that considering only single words as sense disambiguation targets does not allow the SMT system to reliably exploit WSD predictions. This holds even when only targets that meet conditions that are closer to Senseval evaluations, where our WSD models are known to achieve good performance.

8 Multi-word PSD consistently improves translation quality

In contrast with the unreliable single-word WSD results, using phrasal multi-word PSD predictions in SMT remarkably yields better translation quality on *all* test sets, as measured by *all eight* commonly used automatic evaluation metrics. The results are shown in Table 2 for IWSLT and Table 3 for the NIST task. Paired bootstrap resampling shows that the improvements on the NIST test set are statistically significant at the 95% level.

Comparison of the 1-Best decoder output with and without the PSD feature shows that the sentences differ by one or more token respectively for 25.49%, 30.40% and 29.25% of IWSLT test sets 1, 2 and 3, and 95.74% of the NIST test set.

9 Multi-word PSD helps the decoder find a more useful segmentation of the input sentence

Analysis reveals that integrating PSD into SMT helps the decoder select a phrase segmentation of the input sentence which allows to find better translations than word-based WSD. We sampled translation examples from the IWSLT test sets, so that both word-based and phrase-based results are available for comparison. In addition, the relatively short sentence length of this corpus helps give a clearer understanding of the impact of WSD. Consider the following example:

Input 我 想 再 确认 一下 这 张 票 的 预订 。

Reference I want to reconfirm this ticket.

Table 3: Evaluation results on the NIST test set: integrating the PSD translation predictions improves BLEU, NIST, METEOR, WER, PER, CDER and TER.

Experiment	BLEU	NIST	METEOR	METEOR (no syn)	TER	WER	PER	CDER
Baseline	20.20	7.198	59.45	56.05	75.59	87.61	60.86	72.06
+PSD	**20.62**	**7.538**	**59.99**	**56.38**	**72.53**	**85.09**	**58.62**	**68.54**

WSD I would like to reconfirm a flight for this ticket.

PSD I want to reconfirm my reservation for this ticket.

Here, in the input segment “这张票的预订”, the particle “的” is in the same segment as the preceding word when using multi-word PSD predictions (“ 票的 ”), while the single-word WSD prefers to use “ 的预订”. This results in an incorrect translation of the phrase “的预订” as “flight for”. In contrast, PSD prefers to use the target “预订”, which ranks the correct “reservation” as the top translation candidate with a very confident probability of 0.94, as opposed to 0.28 only for the baseline context-independent translation probability used in the single-word WSD-augmented model. Similarly, consider:

Input 请 转 乘 中央 线 。

Reference You should transfer to the Central Line.

WSD Please turn to the Central Line.

PSD Please transfer to Central Line.

Here, PSD translates the segment “转乘” as a single unit and selects the correct translation “transfer to”, while WSD separately translates the words “转” and “乘” into the incorrect “turn to”. The multi-word PSD model correctly ranks “transfer to” as its translation candidate, but it is interesting to note that all other translation candidates (e.g., “have a connection to”) are better than “turn to”, because the sense disambiguation target phrase itself contains disambiguating information, and is therefore a better lexical choice unit. Consider a further example:

Input 我 想 打电话 到 日本 的 东京 ， 现在 东京 是 几 点 ？

Reference I’d like to call Tokyo, Japan. What time is it now in Tokyo?

WSD I want to make a call to Tokyo, Japan is Tokyo time now?

PSD I want to make a call to Tokyo, Japan what time is it now in Tokyo?

The PSD system translates the phrase “是 几 点” as a single target into “what time is”, with a confident PSD probability of 0.90. This prediction is not used by the WSD-augmented system, because the context-independent baseline translation probabilities prefers the incorrect translation “what time does it” higher than “what time is”, with much less confident scores (0.167 vs. 0.004). As a result, using only WSD predictions leads the words “是” and ”几 点” to be translated separately, and incorrectly.

In contrast, the following example demonstrates how multi-word PSD helps in selecting a mix of both longer and shorter phrases where appropriate:

Input 请 给 我 修理 一下 或 给 我 换 一下 。

Reference Please fix it or exchange it.

WSD Please fix it or I change it for me.

PSD Please give me fix it or exchange it for me.

In particular, by translating the phrase “请给我” as a whole, multi-word PSD avoids the problem caused by the incorrect reordering of the pronoun “I” in single-word WSD. The phrase translation is not optimal, but it is better than the single-word WSD translation, which does not make much sense because of the incorrect reordering. At the same time, the multi-word PSD predictions do not translate the phrase “东京 是 几 点” as a single target, which helps pick the better translation “exchange”.

It is worth noting that using multi-word PSD sometimes yields better lexical choice than single-word WSD even in cases when the same phrase segmentation of the input sentence is arrived at. This is the case in the following examples:

Input 全 是 个人 物品 。

Reference This is all my personal luggage.

WSD Is it all personal effects.

PSD They are all personal effects.

Input 咖啡 和 红茶 ， 您 要 哪个 ？

Reference Which would you like, coffee or tea?

WSD Which would you like, and coffee black tea?

PSD Which would you like, black tea or coffee?

The targets that are translated differently are single words in both sentences, which means that the WSD/PSD predictions are identical in the WSD-augmented SMT and PSD-augmented SMT experiments. However, the translation candidate selected by the decoder differs. In the first example, the WSD/PSD scores incorrectly prefer "and" with a probability of 0.967 to the better "or" translation, which is only given a probability of 0.002. However, the PSD-based translation for the whole sentence is correct, while the WSD-based translation is incorrectly ordered, perhaps letting the language model prefer the phrase "and coffee" which was seen 10 times more in the training set than the correctly ordered "and tea". Although this phenomenon requires more analysis, we suspect that having WSD predictions for every phrase in the SMT lexicon allows to learn better log linear model weights than for word-based WSD predictions.

10 When WSD/PSD predictions go wrong

The following examples show that for some sentences using sense disambiguation, whether single-word WSD or multi-word PSD, occasionally does not help or even hurts translation quality. Consider the following example:

Input 我 要 送 餐 服务 。

Reference Room service, please.

WSD I will take meal service.

PSD I want to eat service.

Here, the single word target "送" is incorrectly translated as "eat" and "meal", while a better translation candidate, "order", is given a lower WSD score. Another problem with this sentence is that the word "服务" is not seen alone during training, but in the collocation "房间 服务", so that "服务" was aligned to "service" only during training, and "room service" is not a translation candidate for "服务" in the SMT phrasal translation lexicon. WSD/PSD can only help to rank the given candidates, and there is nothing they can do when the correct translation is not in the original SMT phrasal translation lexicon.

Similarly, consider the following example:

Input 啊 。 给 我 帐单 。

Reference Uhh. Give me a Tab.

WSD Oh. I have the bill.

PSD Well, let me check.

The incorrect translation of "帐 单 。" as "check ." by the multi-word PSD model inappropriately influences the translation of the context, resulting in a sentence translation whose meaning has nothing in common with the reference.

This, of course, highlights the fact that for extremely short sentences containing only neutral words or extremely polysemous function words, WSD/PSD is not a good idea. In Example 7, there is actually no solid contextual evidence upon which the sense disambiguation model can decide whether "帐 单" should be translated as "bill", "check", or "tab". "给" is the highly polysemous verb "give", and "我" is the neutral word "I". In fact, without document level context, it would be hard even for a human translator to pick the right translation.

These observations suggest that in future evolutions of these directions, we might want to trigger PSD based on a cursory examination of sentence properties, in order to avoid hurting translation quality when there is simply no context information for PSD to exploit.

11 Conclusion

We have presented new comparative empirical evidence and data analysis strongly indicating that in order to be useful for improving the translation quality of current phrasal SMT performance levels, we will need *phrase sense disambiguation* models that are generalized to disambiguate phrasal target words, rather than traditional single-word sense diambiguation models. On one hand, the experimental results conducted on both the IWSLT-06 and NIST Chinese-English translation tasks, using eight different automatic evaluation metrics, have shown that—remarkably—incorporating phrase sense disambiguation *consistently* improves translation quality on *all* test sets for *all* evaluation metrics. But on the other hand, contrastive results where traditional single-word oriented WSD is incorporated into SMT leads to unpredictable effects on translation quality depending on the metric used, thus tending to confirm that the generalization from word sense disambiguation to phrase sense disambiguation is indeed necessary.

Analysis suggests that this very different behavior is made possible by the dynamic integration of phrase-based WSD predictions into SMT, which allow all phrase targets to compete during decoding, instead of forcing the SMT system to use a particular segmentation of its input sentence.

References

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for mt evaluation with improved correlation with human judgement. In *Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics (ACL-2005)*, Ann Arbor, Michigan, June 2005.

Peter Brown, Stephen Della Pietra, Vincent Della Pietra, and Robert Mercer. Word-sense disambiguation using statistical methods. In *29th meeting of the Association for Computational Linguistics*, pages 264–270, Berkeley, California, 1991.

Clara Cabezas and Philip Resnik. Using wsd techniques for lexical selection in statistical machine translation. Technical report, Institute for Advanced Computer Studies, University of Maryland, 2005.

Marine Carpuat and Dekai Wu. Word sense disambiguation vs. statistical machine translation. In *the annual meeting of the association for computational linguistics (ACL-05)*, Ann Arbor, Michigan, 2005.

Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, Prague, June 2007.

Marine Carpuat, Weifeng Su, and Dekai Wu. Augmenting ensemble classifi cation for word sense disambiguation with a Kernel PCA model. In *Senseval-3, Third International Workshop on Evaluating Word Sense Disambiguation Systems*, Barcelona, July 2004. SIGLEX, Association for Computational Linguistics.

Marine Carpuat, Yihai Shen, Xiaofeng Yu, and Dekai Wu. Toward integrating word sense and entity disambiguation into statistical machine translation. In *Third International Workshop on Spoken Language Translation (IWSLT 2006)*, Kyoto, November 2006.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. Word sense disambiguation improves statistical machine translation. In *45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, Prague, June 2007.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *the 43th Annual Meeting of Computational Linguistics (ACL-2005)*, Ann Arbor, Michigan, June 2005.

Timothy Chklovski, Rada Mihalcea, Ted Pedersen, and Amruta Purandare. The senseval-3 multilingual english-hindi lexical sample task. In *Senseval-3, Third International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 5–8, Barcelona, Spain, July 2004. SIGLEX, Association for Computational Linguistics.

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *the Human Language Technology conference (HLT-2002)*, San Diego, CA, 2002.

Ismael Garcia-Varea, Franz Och, Hermann Ney, and Francisco Casacuberta. Refi ned lexicon models for statistical machine translation using a maximum entropy approach. In *the 39th annual meeting of the association for computational linguistics (ACL-01)*, Toulouse, France, 2001.

Ismael Garcia-Varea, Franz Och, Hermann Ney, and Francisco Casacuberta. Effi cient integration of maximum entropy lexicon models within the training of statistical alignment models. In *AMTA-2002*, pages 54–63, Tiburon, California, October 2002.

Jesús Giménez and Lluís Màrquez. Context-aware discriminative phrase selection for statistical machine translation. In *Workshop on Statistical Machine Translation*, Prague, June 2007.

Adam Kilgarriff and Joseph Rosenzweig. Framework and results for english senseval. *Computers and the Humanities*, 34(1):15–48, 1999. Special issue on SENSEVAL.

Adam Kilgarriff. English lexical sample task description. In *Senseval-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 17–20, Toulouse, France, July 2001. SIGLEX, Association for Computational Linguistics.

Philipp Koehn. Pharaoh: a beam search decoder forphrase-based statistical machine translation models. In *6th Conference of the Association for Machine Translation in the Americas (AMTA)*, Washington, DC, September 2004.

Gregor Leusch, Nicola Ueffi ng, and Hermann Ney. Effi cient mt evaluation using block movements. In *EACL-2006 (11th Conference of the European Chapter of the Association for Computational Linguistics)*, pages 241–248, Trento, Italy, April 2006.

Hwee Tou Ng and Yee Seng Chan. English lexical sample task via english-chinese parallel text. In *4th International Workshop on Semantic Evaluation (SemEval-2007)*, Prague, June 2007.

Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, 2004.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *AMTA*, pages 223–231, Boston, MA, 2006. Association for Machine Translation in the Americas.

Lucia Specia, Maria das Gračas Volpe Nunes, and Mark Stevenson. Learning expressive models for word sense disambiguation. In *45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, Prague, June 2007.

Christoph Tillmann, Stefan Vogel, Hermann Ney, A. Zubiaga, and H. Sawaf. Accelerated dp-based search for statistical translation. In *Eurospeech'97*, pages 2667–2670, Rhodes, Greece, 1997.

David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. Word-sense disambiguation for machine translation. In *Joint Human Language Technology conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver, 2005.

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, 1997.

Demonstration of the Dutch-to-English METIS-II MT System

Peter Dirix, Vincent Vandeghinste, and Ineke Schuurman
Centre for Computational Linguistics
Katholieke Universiteit Leuven, Belgium
{peter,vincent,ineke}@ccl.kuleuven.be

1 Introduction

The European METIS-II project[1] (Oct. 2004-Sept. 2007) combines techniques from rule-based and corpus-based MT in a hybrid approach for four language pairs (German, Dutch, Spanish, and Greek to English). We only use a dictionary, basic analytical resources and a monolingual target-language corpus in order to enable the construction of an MT system for lesser-resourced languages. Cutting up sentences in linguistically sound subunits improves the quality of the translation. Demarcating clauses, verb groups, noun phrases, and prepositional phrases restricts the number of possible translations and hence also the search space. Sentence chunks are translated using a dictionary and a limited set of mapping rules. Using bottom-up matching to match the different translated items and higher-level structures with the database information, one or more candidate translations are constructed. A search engine ranks them using occurence frequencies and match accuracy in the target-language corpus.

2 Components

The source-language analysis tools construct a source-language model. This toolset consists of a tokeniser, the TnT tagger trained on the Spoken Dutch corpus, a PoS-based lemmatiser, a chunker, and a subclause delimiter.

The translation model consists of a bilingual Dutch-English dictionary with approximately 110,000 entries and a set of tag-mapping rules between Dutch and English.

The target-language model is based on a target-language corpus, the British National Corpus (BNC). It is processed in an analogous way to the source-language input sentences. The translation engine itself is composed of an expander and a ranker. The expander inserts, deletes, moves and permutes tokens and chunks generated during dictionary look-up and the application of the tag mapping. There are currently some half a dozen rules applying. The ranker is a beam-search, bottom-up algorithm that ranks the proposed translations according to the language model. It does not alter the translations anymore. Finally, a token generator generates the correct word forms, since in all intermediate processes, only lemmas are used.

More information on the different components of the system can be found in (Dirix et al., 2005), (Dirix et al., 2006), and (Vandeghinste et al., 2006). The impact of applying hand-crafted rules is described in (Vandeghinste et al., 2007).

3 Evaluation

Our test set consists of 50 Dutch sentences, selected from newspaper texts, with three human reference translations. These sentences are selected to contain a number of classical difficult MT issues. The system generates several translation alternatives (dependent on beam size, which is 20 for all tests described in this paper), each with a weight. As our sys-

[1]Supported by the 6th European Framework Programme, FP6-IST-003768.

tem is not always capable of generating only one best translation, we present two types of results, namely the average BLEU scores of all the top-weight[2] translations generated for that test sentence ('average' score) and the highest BLEU scores of all the top-weight translations generated for that test sentence ('best' score).

Table 1: BLEU scores

	BLEU
'average'	0.3024
'best'	0.3486

A discussion of the results in Table 1 can be found in (Vandeghinste et al., 2007).

4 Current and future work

Currently, we are adding co-occurrence metrics in order to generate unique top-weight translations. These metrics are used to differentiate the weights of the different translations of a single source-language dictionary entry. It is based on the co-occurence of the differents words of the sentence in the target-language corpus. We also moved to an xml representation of our dictionary in order to better represent complex entities. We allow structural changes and discontinuous entries.

Furthermore, we are developing a post-editing interface. The corrections of human post-editors will result in an aligned corpus of machine-made and corrected translations. The corrected translations can be added to the target-language corpus and will also be used as part of the bilingual dictionary. This can be seen as a kind of supervised machine learning.

5 Related work

Related techniques are context-based machine translation (CBMT), as described in (Carbonell et al., 2006), and generation-heavy hybrid machine translation (GHMT), as described in (Habash, 2003). As in METIS, CBMT does not rely on parallel corpora, but on a large target-language corpus, an optional small source-language corpus and a bilingual dictionary. The translation and target-language generation phases do not require any linguistic knowledge, but use n-grams instead. GHMT uses about the same resources as CBMT, but involves a deep source-language analysis. Initially, the dependency structure of the source language is maintained, but at the end, a source-language-independent generation module rewrites the target language part lexically and syntactically.

[2]The *top-weight* translations are those translations that receive the highest weight.

References

Jaime Carbonell, Steve Klein, David Miller, Michael Steinbaum, Tomer Grassiany, and Jochen Frey, 2006. *Context-Based Machine Translation.* In *MTA 2006: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, "Visions for the Future of Machine Translation"*, pp. 19–28.

Peter Dirix, Vincent Vandeghinste, and Ineke Schuurman, 2005. *METIS-II: Example-based translation using monolingual corpora – System description.* In *Proceedings of MT Summit X, Workshop on EBMT*, pp. 43–50.

Peter Dirix, Vincent Vandeghinste, and Ineke Schuurman, 2006. *A new hybrid approach enabling MT for languages with little resources.* In *Proceedings of the 16th Meeting of Computational Linguistics in the Netherlands*, pp. 117–132.

Nizar Habash, 2003. *Matador: a large-scale Spanish-English GHMT system.* In *Proceedings of MT Summit IX*, pp. 149–156.

Vincent Vandeghinste, Ineke Schuurman, Michael Carl, Stella Markantonatou, and Tony Badia, 2006. *METIS-II: Machine Translation for Low Resource Languages.* In *Proceedings of the 5th International Confererence on Language Resources and Evaluation (LREC).*

Vincent Vandeghinste, Peter Dirix, and Ineke Schuurman, 2007. *The effect of a few rules on a data-driven MT system.* In *Proceedings of the METIS-II Workshop: New Approaches to Machine Translation*, pp. 27–34.

A New Method for the Study of Correlations between MT Evaluation Metrics

Paula Estrella
ISSCO/TIM/ETI
University of Geneva
40, bd. du Pont-d'Arve
1211 Geneva, Switzerland
paula.estrella@issco.unige.ch

Andrei Popescu-Belis
ISSCO/TIM/ETI
University of Geneva
40, bd. du Pont-d'Arve
1211 Geneva, Switzerland
andrei.popescu-belis@issco.unige.ch

Maghi King
ISSCO/TIM/ETI
University of Geneva
40, bd. du Pont-d'Arve
1211 Geneva, Switzerland
Maghi.King@gmail.com

Abstract

This paper aims at providing a reliable method for measuring the correlations between different scores of evaluation metrics applied to machine translated texts. A series of examples from recent MT evaluation experiments are first discussed, including results and data from the recent French MT evaluation campaign, CESTA, which is used here. To compute correlation, a set of 1,500 samples for each system and each evaluation metric are created using bootstrapping. Correlations between metrics, both automatic and applied by human judges, are then computed over these samples. The results confirm the previously observed correlations between some automatic metrics, but also indicate a lack of correlation between human and automatic metrics on the CESTA data, which raises a number of questions regarding their validity. In addition, the roles of the corpus size and of the selection procedure for bootstrapping (low vs. high scores) are also examined.

1 Introduction

One of the design principles of automatic MT evaluation metrics is that their scores must "correlate" with a reliable measure of translation quality, generally estimated by human judges. Indeed, the claim that an automatic scoring procedure applied to MT output can provide an accurate view of translation quality must be substantiated by a proof that the scores do reflect genuine quality, as perceived by human users of a translation. For instance, the proponents of BLEU or WNM (Babych and Hartley, 2004; Papineni et al., 2001) have compared the scores produced by their metrics – which compare n-grams of MT-generated sentences with one or more reference translations produced by humans – with adequacy and fluency scores assigned by human judges.

It is not, of course, that all metrics of translation quality *must* be correlated. Although adequacy (i.e. fidelity or "semantic correctness") and fluency (acceptability as a valid sample of the target language) do seem correlated to some extent (White, 2001), one can easily imagine MT output with high fluency but low adequacy. However, an automatic MT evaluation metric should at least correlate with one quality characteristic on which human judges would reliably agree, which can be some aspect of intrinsic quality, or a utility-based measure with respect to a given task.

Given the low cost of automatic metrics, they have been widely used in recent experiments, three of which are discussed in Section 5. However, the results obtained on the correlation between metrics that were used are difficult to compare, and therefore the reliability of automatic metrics is hard to assess.

In this article, we propose a method to measure the correlation between two MT evaluation metrics based on bootstrapping (Section 3) and apply it to data from the recent French MT evaluation campaign, CESTA

(Section 4). Our experiments (Section 5) analyze the correlation between metrics and show that correlation is lower than expected for automatic *vs*. human metrics. The experiments also show that correlation varies with sample size, as well as with the subset of sentences that is considered (low vs. high quality). Samples from the two CESTA runs indicate however that correlations do not vary significantly with a different translation domain.

2 Correlation between MT Evaluation Metrics in Previous Experiments

Many authors report on the correlation between human and automated metrics: some working at the sentence level (Kulesza and Shieber, 2004; Russo-Lassner et al., 2005), and some at the corpus level (Doddington, 2002; Papineni, 2002), in a variety of approaches and setups. Recent experiments, for instance, report that the correlation of the well-known BLEU metric with metrics applied by humans is not always as high as previously reported (Callison-Burch et al., 2006). In this section, we analyze three recent contributions that illustrate clearly the variety of methodologies used to compute correlations between metrics.

2.1 An Experiment with the Europarl Corpus

Koehn and Monz (2006) describe the competition organized during the Statistical MT Workshop at NAACL 2006. Its main goal was to establish baseline performance of MT evaluation for specific training scenarios. The test corpus consisted of sentences from the Europarl corpus (Koehn, 2005) and from editorials of the Project Syndicate website, and contained a total of 3,064 sentences. The translation directions were SP↔EN, FR↔EN, DE↔EN and there were 14 participating systems.

The BLEU metric was used for automatic evaluation, as the most commonly used metric in the MT community. To provide human quality judgments, the workshop participants had to assess 300–400 sentences each, in terms of adequacy and fluency, on a 5-point scale. Each evaluator was in fact simultaneously given 5 machine translations, one reference translation, and one source sentence, and was asked to perform a comparative evaluation of the machine translations. The scores for adequacy and fluency were then normalized and were finally converted into rankings, to increase robustness of the conclusions.

The similarity between the performances of the systems and the problems encountered in the human evaluation made it difficult to draw strong conclusions about the correlation of human and automatic metrics. Some evaluators explicitly pointed out how difficult it was to maintain consistency of judgment, especially when the sentences are longer than average. Evaluators also suggested extending the scale for adequacy scores, as this would improve the reliability of judgments.

2.2 Reliability and Size of Test Set

Coughlin (2003) reports results on the correlation between human assessments of MT quality and the BLEU and NIST metrics (Doddington, 2002) in a large scale evaluation, using data collected during two years. The judges were neither domain experts (in computer science), nor were they involved in the development of the participating systems. Having access only to high quality reference translations, they had to rate sentences in pairs, to compare two different systems. The innovative methodology of human evaluation was to rate the overall *acceptability* of the sentences – and not their adequacy or fluency – on a 4-point scale, without further instructions, thus generating only one human score per sentence.

The sentences were evaluated by 4–7 judges, leading to an average inter-rater agreement of 0.76 for EN→DE and 0.83 for FR→EN.

Contrary to the work described in the previous subsection, Coughlin (2003) found a very high correlation between the BLEU metric and the human judges, especially when test data sets comprise more than 500 sentences. For the NIST metric, on the contrary, correlation is lower for data sets that comprise more than 250 sentences. In general, Coughlin (2003) shows a high correlation between BLEU/NIST and human scores, for all language pairs and systems used, except for the FR→EN pair which had low negative correlation, for which they suggest that the Hansard domain might be more difficult to translate for the systems under evaluation.

2.3 Correlations in the CESTA Campaign

The French MT evaluation campaign, CESTA, also reported results on the meta-evaluation of automatic metrics, i.e. their comparison to the human scores of adequacy and fluency (Hamon et al., 2006). The data used for the evaluation is described in detail in Section 4, since it is also used in this paper. The main automatic metrics used in CESTA are BLEU, NIST, Weighted N-gram Metric (WNM) (Babych, 2004), mWER (Niessen et al., 2000), and mPER (Tillmann et al., 1997).

CESTA used human judges to assign adequacy and fluency scores on a 5-point scale with a protocol and interfaces that changed from the first to the second run. The rating scale in the first run explicitly listed the intermediate labels for the values, while for the second run the labels were removed. In addition, while in the first run the evaluation of adequacy and fluency was done at the same time, in the second run, the judges scored every segment separately for fluency and for adequacy. In both runs the final scores for each sentence are the average of two assessments.

When defined as the percentage of identical values from the 5-point scale, the inter-judge agreement is only 40% for fluency, and varies from 36% to 47% for adequacy in the first vs. second run (EN→FR). However, when defined as the percentage of scores that differ by at most one point between two judges (e.g. a segment rated 3 by one judge and 2 by the other would count as an agreement), inter-judge agreement increases significantly, to 84% for fluency and 78% for adequacy. Moreover, the CESTA campaign reports acceptable correlation between automatic metrics and adequacy/fluency, when computed over the five participating systems, that is, as the Pearson correlation of five pairs of values. For example, the correlation of NIST (or BLEU) with fluency is around 0.67 in the first run[1].

3 Using Bootstrapping to Study the Correlation between Metrics

We propose here the use of bootstrapping to investigate the correlation between the scores of different metrics on a *per system* basis, and not only between the various systems participating in an evaluation. To calculate the correlation between two or more variables (metrics in this case), we need two or more samples of each variable: for example, in an evaluation campaign, the samples are the final scores obtained by each system, which are then correlated to explore relations between different metrics (cross-system correlation). Our approach consists of (artificially) generating several sample scores of the same system and calculating the correlations of two metrics over the set of samples, for that particular system. The advantages of this method are that we only need the output of one system and that the results obtained are specific to that system. The disadvantage is of course, that direct comparison with standard cross-system correlation is not possible, since we only consider one system at a time.

Therefore, this method can be used to estimate the correlation of metrics as the result of evaluating one system only, and can include of course any kind of metrics, human and automatic, in the analysis.

3.1 Bootstrapping Samples of Scores

Bootstrapping is a statistical technique that is used to study the distribution of a variable based on an existing set of values (Efron and Tibshirani, 1993). This is done by randomly resampling *with replacement* (i.e. allowing repetition of the values) from the full existing sample and computing the desired parameters of the distribution of the samples. The method has the practical advantage of being easy to implement and the theoretical advantage of not presupposing anything about the underlying distribution of the variable. A simple programming routine can thus calculate the estimators of the mean, variance, etc., of any random variable distribution.

Moreover, when the original sample is resampled a large number of times, the law of large numbers ensures that the observed probability approaches (almost certainly) the actual probability. Also, when N is sufficiently large, the sample scores are quite close to the normal distribution, as illustrated in Figure 1.

The bootstrapping algorithm can be summarized as follows:

[1]The CESTA final report provides the detailed scores: http://technolangue.net/IMG/pdf/Rapport_final_CESTA_v1.04.pdf.

1. Given a sample $X = (X_1, X_2, \ldots, X_n)$ from a population **P**, generate N random samples (noted X^*) of the same size by drawing n values from the sample, with replacement (each value having probability $1/N$).
2. The resulting population $\mathbf{P}^*$, noted $X^* = (X_1^*, \ldots, X_N^*)$, constitutes the N bootstrapped samples.
3. If the original estimator of a given population parameter was $\theta(X)$, with the bootstrapped samples we can calculate the same estimator as $\theta(X^*)$.

An important parameter for bootstrapping is N, the number of bootstrapped samples, i.e. the number of times the process is repeated. This number should be large enough to build a representative number of samples. It appears that, for instance, $N = 200$ leads to slightly biased estimations (Efron and Gong, 1983; Efron and Tibshirani, 1993; Koehn, 2004; Zhang et al., 2004, so N ~ 1,000 is preferred, for example N = 1,000) or even $N = 10{,}000$ (Bisani and Ney, 2004). Based on these examples, we decided to use $N = 1{,}500$ bootstrapped samples.

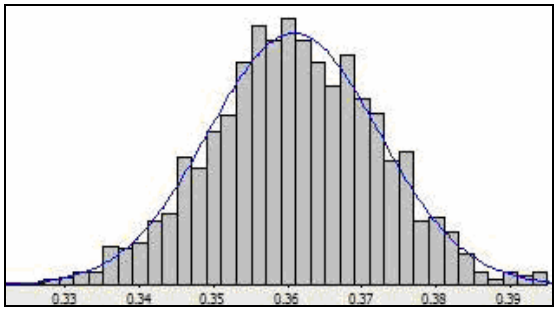


Figure 1. Example of histogram for the WER scores obtained with 1,500 bootstrapped samples (CESTA scores, first run, system S2)

3.2 Application to MT Evaluation Scores

In the MT field, bootstrapping has been mainly used to estimate confidence intervals for automatic metrics and to compute the statistical significance of comparative performance of different MT systems, e.g. using the BLEU (Koehn, 2004; Kumar and Byrne, 2004; Zhang et al., 2004) or WER metric (Bisani and Ney, 2004). Here, bootstrapping will be used to compute the correlation between metrics for MT. These correlations will be studied for each system, i.e. they are calculated on a *per system* basis as opposed to the common cross-system correlation.

Since correlation concerns two sets of scores, we need to apply the metrics simultaneously to the same bootstrapped samples to keep consistency in the scores. Put in simpler words, we apply two (or more) different metrics to the same random sample per iteration of the bootstrapping process. A *random sample* is a set of segments randomly selected from the corpus and of the same size of the corpus used in the evaluation.

Described in pseudo code, the routine computing correlation is particularly simple: M is the number of segments to be considered, N is the numbers of iterations, `sample[m]` is the m-th element of the random sample and `sample*` is the complete bootstrapped sample:

```
for(n=0; n<N; n++){
    for(m=0; m<M; m++){
       sample[m] = selectRandSeg();
    }
    scoresA[n]=calcMetricA(sample*);
    scoresB[n]=calcMetricB(sample*);
}
   calcCorrelation(scoresA, scoresB);
```

4 Evaluation Resources: Data, Systems and Metrics

For the experiments presented here, we used the resources of the EN→FR translation task in the CESTA MT evaluation campaign (Hamon et al., 2006). In all cases, the results of the participating systems are anonymized, therefore the systems will simply be referred to by the codes S1 to S5 in no particular order.

One of the goals of the first run was to validate the use of automatic evaluation metrics with French as a target language, by comparing the results of well-known automatic metrics with fluency and adequacy scores assigned by human judges. The test data for the first run consisted of 15 documents from the Official Journal of the European Communities (JOC, 1993) with a total of 790 segments and an average of 25 words per segment. The documents contain transcribed questions and answers in a parliamentary context, and since no particular domain was

targeted when putting together the corpus, the CESTA campaign considered this as *general domain* data. Five systems participated in the EN→FR first run, both commercial and research ones.

For the second run, the goal was to improve the evaluation protocols used in the first run and to observe the impact of system adaptation to a particular domain. Therefore, the *medical domain* was chosen, using data collected from the *Santé Canada* website, with a total of 288 segments and an average of 22 words per segment. Almost the same systems participated in the second run.

In addition to the automatic metrics used in the CESTA campaign, we included in our experiment precision and recall from the General Text Matcher (Turian et al., 2003).

5 Experimental Study of Correlation

Although we performed the study using all the systems participating in the CESTA campaign, we will only present here the results of two systems, namely S2 and S5, chosen among the best. In Section 5.1, we compute correlations between metrics on two test sets of dissimilar size, in Section 5.2 we study the correlations for segments of very high and very low adequacy scores and, finally, in Section 5.3 we present the results of the correlations for a test set of a different domain.

5.1 Correlation Values and the Influence of the Size of Test Data

In the first experiment, we compared correlation between metrics, when calculated on a test set of 5 documents and on a larger set of 15 documents from the general domain corpus. We hypothesize that if a strong correlation exists between two score sets, it should be stable, i.e. it should be similar or even higher, when using a larger test set.

Tables 1 to 4 show the Pearson R coefficients for all the metrics applied in this study, separately for systems S5 (Tables 1 and 2) and S2 (Tables 3 and 4). The correlation figures were computed on 5 documents in Tables 1 and 3, and respectively on 15 documents in Tables 2 and 4. Negative values generally occur when the metrics vary in the opposite direction, e.g. higher scores of the first one correspond (correctly) to lower scores of the second one.

As we expected, there is a relatively high correlation between metrics of the same type (except for adequacy and fluency for S5) regardless of the size of the test data set: for instance, the following correlations between metrics appear to be quite high: WER *vs*. PER > 0.81, BLEU *vs*. NIST > 0.72, PREC *vs*. REC > 0.76. However, the figures show also that automatic metrics correlate better with other automatic metrics than with adequacy or fluency; for both systems, the NIST metric presents the lowest coefficients.

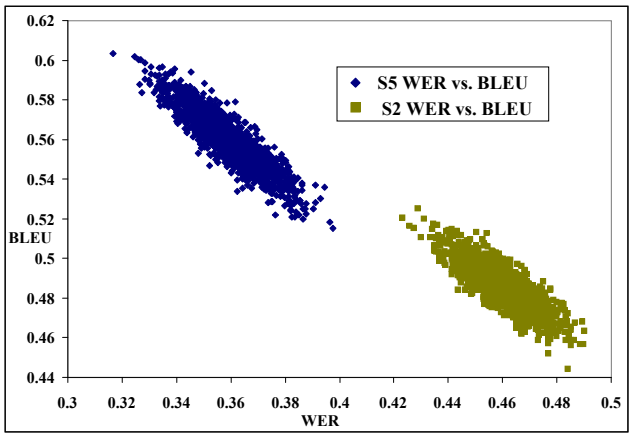


Figure 2. Scatter plot of WER vs. BLEU bootstrapped scores using 5 documents

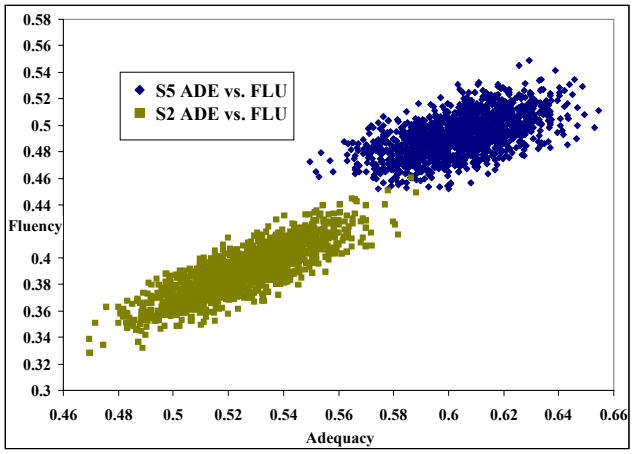


Figure 3. Scatter plot of adequacy vs. fluency bootstrapped scores using 5 documents

Regarding the change in the size of the test data, the correlations (excluding adequacy *vs*. fluency) for S2 systematically increase when using 15 documents with respect to 5. However, this is less clear for S5: the correlation of NIST

with all other metrics increases, BLEU *vs*. WER/PER remains stable, but the correlations between automatic metrics and the human ones decrease, quite considerably in some cases, e.g. BLEU *vs*. fluency. This is probably due to the particular documents selected, since scores vary more on small test sets, as shown in (Estrella et al., 2007).

A graphical representation of the scores appears in Figures 2 to 5, which plot two scores for each of the 1,500 bootstrapped samples, for systems S2 (light/green) and S5 (dark/blue). Figure 2 illustrates two metrics that are highly correlated, BLEU and WER: the clouds of dots are organized along a line, which has negative slope as lower WER corresponds to higher BLEU (and to better performance, in principle). The correlation coefficients for the samples in Figure 2 are respectively -0.83 and -0.89.

A similar, albeit lower, correlation appears in Figure 3 for the two human metrics, adequacy vs. fluency. Again, the clouds of dots are organized along lines, this time with positive slopes. The correlation coefficients are respectively 0.84 and 0.58 for S2 and S5, the lower value for S5 being quite visibly reflected in the more scattered pattern of blue dots (less linear and more rounded shape).

S5	WER	PER	BLEU	NIST	ADE	FLU	PREC	REC
WER	1	0.93	-0.90	-0.69	-0.42	-0.43	-0.72	-0.56
PER		1	-0.89	-0.76	-0.40	-0.41	-0.84	-0.68
BLEU			1	0.83	0.39	0.44	0.82	0.71
NIST				1	0.26	0.27	0.87	0.68
ADE					1	0.58	0.34	0.39
FLU						1	0.34	0.37
PREC							1	0.79
REC								1

Table 1. Correlation matrix for S5 using 5 documents

S5	WER	PER	BLEU	NIST	ADE	FLU	PREC	REC
WER	1	0.92	-0.90	-0.75	-0.28	-0.32	-0.74	-0.55
PER		1	-0.89	-0.79	-0.25	-0.29	-0.84	-0.65
BLEU			1	0.86	0.25	0.29	0.83	0.66
NIST				1	0.16	0.16	0.86	0.64
ADE					1	0.63	0.25	0.30
FLU						1	0.24	0.26
PREC							1	0.78
REC								1

Table 2. Correlation matrix for S5 using 15 documents

S2	WER	PER	BLEU	NIST	ADE	FLU	PREC	REC
WER	1	0.81	-0.83	-0.52	-0.48	-0.46	-0.61	-0.41
PER		1	-0.73	-0.60	-0.43	-0.42	-0.75	-0.54
BLEU			1	0.72	0.43	0.41	0.74	0.61
NIST				1	0.13	0.13	0.84	0.58
ADE					1	0.84	0.27	0.32
FLU						1	0.26	0.30
PREC							1	0.76
REC								1

Table 3. Correlation matrix for S2 using 5 documents

S2	WER	PER	BLEU	NIST	ADE	FLU	PREC	REC
WER	1	0.83	-0.85	-0.59	-0.49	-0.49	-0.64	-0.50
PER		1	-0.81	-0.69	-0.44	-0.43	-0.79	-0.61
BLEU			1	0.79	0.43	0.43	0.78	0.65
NIST				1	0.23	0.20	0.86	0.61
ADE					1	0.79	0.30	0.35
FLU						1	0.28	0.33
PREC							1	0.77
REC								1

Table 4. Correlation matrix for S2 using 15 documents

5.2 Correlation for High and Low Quality Translations

The findings from the previous section can be due to many factors; for example, using a corpus containing segments of diverse translation difficulty or using the average of two judgments for adequacy or fluency might give less informative results, since the final scores are calculated on the entire test set. Or it might be, as pointed out by Coughlin (2003), that humans could be influenced by the reference translation they see during the evaluation and therefore evaluate systems depending more on the algorithm they use (statistical or rule-based) than on their intrinsic quality.

To further investigate the correlations described in Sections 5.1, we carried out another experiment, focusing on the highest and lowest scores assigned by adequacy judgments. The goal is to explore the agreement among some metrics when the adequacy scores are very high and very low. An *a priori* hypothesis is that low quality translations might be more difficult to evaluate (leading to a larger variation of scores) than high quality translations. According to this hypothesis, the correlation between metrics applied on almost perfect segments should be stronger than that of metrics applied on low quality segments. We consider "quality" in terms of the score provided by human judges of adequacy, fluency or the average of both; for the purpose of this experiment we take adequacy as the measure of quality, but results using fluency or the average do not change dramatically.

Each segment of the CESTA data was evaluated for adequacy and for fluency by two judges, and the final scores for each metric are the average between the two assessments. These scores were then normalized and converted from a 5-point scale to a value between 0 and 1. To find only the segments with high adequacy score, we extracted, from the 15 documents of the first run, those segments with an average adequacy score above 0.825. For the low quality test set, we extracted the segments with an average adequacy below 0.125. We tried to keep the size constant, so we had around 130 segments in both new test sets, given that S5 had the least number of segments below 0.125. These empirical cut-off limits should also account for high inter-judge agreement, since a high/low score can only be reached if both assessors assigned similar high/low scores for the same segment.

To simplify the experiment, we only applied the WER and PER metrics to the corresponding outputs of S2 and S5. Tables 5 and 6 show the resulting R coefficients, the lower part of the tables corresponding to S2 and the upper part to S5 (for compactness reasons).

S2 \ S5	WER	PER	FLU	ADE
WER		0.93	-0.17	-0.25
PER	0.71		-0.13	-0.28
FLU	-0.14	-0.11		*-0.13*
ADE	-0.09	-0.14	0.16	

Table 5. Correlations on the low-adequacy data set: S2 lower-left half, S5 upper-right

S2 \ S5	WER	PER	FLU	ADE
WER		0.94	-0.17	-0.32
PER	0.93		-0.27	*-0.10*
FLU	-0.43	-0.39		0.42
ADE	-0.36	-0.30	0.41	

Table 6. Correlation on the high-adequacy data set: S2 lower-left half, S5 upper-right

The correlations clearly increase in absolute value from low-adequacy to high-adequacy segments, as hypothesized, but are still much weaker than expected for high-adequacy segments. Two special cases with extremely low correlation values are marked in italics, namely fluency *vs*. adequacy in Table 5 and PER *vs*. adequacy in Table 6, respectively. In the first case, we manually inspected the results of the bootstrapping procedure, and observed that adequacy scores were much lower than the fluency scores. Figures 4 and 5 provide a graphical representation of these two cases.

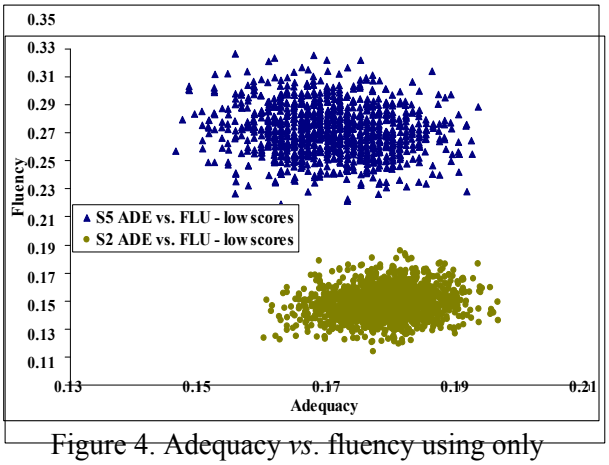


Figure 4. Adequacy *vs*. fluency using only segments with low adequacy scores

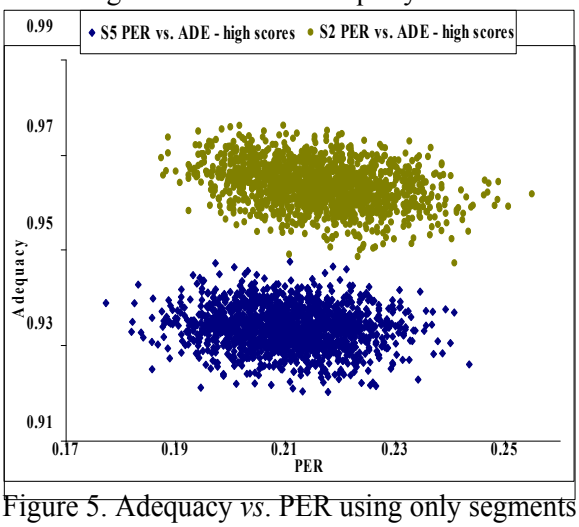


Figure 5. Adequacy *vs*. PER using only segments with high adequacy scores

For the PER *vs*. adequacy correlation, we found out that S2 has more segments scoring less than 0.125 (90 segments *vs*. 57 for S5) but has also more segments scoring 1 (121 segments vs. 93 for S5). This explains the scatter plot in Figure 5 but contradicts the expected results, since S5 was ranked among the best in the CESTA campaign. In overall scores this situation could be changed because the scores are averaged out. In practice, we believe that the difference between coefficients of -0.10 and -0.13 does not have a big impact, since one system provides clearly better translations than the other.

5.3 Correlations on a Different Domain

The last experiment consists of comparing the correlations obtained for test sets in a different domain than the previous one. For the second run of the CESTA campaign, the participants had the opportunity to train or adapt their systems to a particular domain (medical) using a special corpus for that purpose. Given that systems were trained for that specific domain, performance should have increased, as well as correlations between some metrics. Using the test corpus created for the second run of CESTA (288 segments), the results are comparable, in terms of size, to those obtained in Section 5.1 for 5 documents (270 segments).

Results for S2 an S5 are reported respectively in Tables 7 and 8. For the human metrics, results are not directly comparable to those of the previous sections due to a change in the evaluation protocols from the first run of the campaign to the next. Unfortunately, it appears that correlation coefficients remain quite low, despite the adaptation. In Table 7 we observe a significant increase in correlation coefficients between automatic metrics and adequacy for S2; this difference between S5 and S2 might indicate a failure of S5 to fully acquire the relevant vocabulary for the new domain. Following the hypothesis of the previous section and recalling that S2 was ranked below S5 in the CESTA campaign, it appears that assessment of low quality segments leads to more variation of scores, thus resulting in low correlation coefficients.

S2	WER	PER	BLEU	NIST	ADE	FLU	PREC	REC
WER	1	0.98	-0.87	-0.72	-0.72	-0.27	-0.69	-0.77
PER		1	-0.81	-0.69	-0.70	-0.26	-0.67	-0.83
BLEU			1	0.84	0.68	0.36	0.77	0.47
NIST				1	0.51	0.24	0.68	0.40
ADE					1	0.27	0.50	0.52
FLU						1	0.27	0.15
PREC							1	0.35
REC								1

Table 7. Correlation matrix for S2 using corpus from health domain

S5	WER	PER	BLEU	NIST	ADE	FLU	PREC	REC
WER	1	0.87	-0.82	-0.67	-0.20	-0.28	-0.66	-0.29
PER		1	-0.80	-0.75	-0.18	-0.20	-0.78	-0.44
BLEU			1	0.80	0.17	0.21	0.74	0.48
NIST				1	0.21	0.21	0.85	0.63
ADE					1	0.34	0.18	0.13
FLU						1	0.15	0.12
PREC							1	0.64
REC								1

Table 8. Correlation matrix for S5 using corpus from health domain

6 Conclusion and Future Work

The method presented in this paper allows the computation of correlation between two metrics on a single system, using bootstrapping to create a large set of samples of variable qualities.

Observations clearly indicate that some related automatic metrics, such as BLEU and NIST, or BLEU and WER, are better correlated than automatic vs. human metrics. However, even for related metrics, the correlation is not necessarily very high.

It is quite surprising that, using this method, correlations between human and automatic metrics are much lower than figures obtained by other methods and published as arguments for the reliability of automatic metrics.

At this stage, it is not yet clear, which is the main factor that explains such a low correlation, and whether these figures do indicate a significant *lack of correlation* on the CESTA scores that we examined. For instance, these figures could be related to low inter-rater agreement between the two judges of adequacy and fluency, which is not compensated by the use of the average values or to the fact that these automatic metrics are not suitable for the evaluation of morphologically richer languages, such as French.

Future work in this direction will examine how human scores used in our experiments are distributed among systems. Of course, adding new human judgments of the same MT output could help to increase our confidence in adequacy and fluency, but this operation is quite costly. We also plan to repeat some of the experiments with other automatic metrics, which claim to improve some of the metrics used here and to improve correlation with human scores.

References

Babych, B. 2004. Weighted N-gram model for evaluating Machine Translation output. In *CLUK 2004*. Birmingham, UK.

Babych, B., and T. Hartley 2004. Extending the BLEU MT Evaluation Method with Frequency Weightings. In *ACL 2004 (42nd Annual Meeting of the Association for Computational Linguistics: ACL 2004)*, 621-628. Barcelona, Spain.

Bisani, M., and H. Ney 2004. Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. In *IEEE International Conference on*

Acoustics, Speech, and Signal Processing, 409-412. Montreal, Canada.

Callison-Burch, C., M. Osborne, and P. Koehn 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In *EACL 2006 (11th Conference of the European Chapter of the Association for Computational Linguistics)*, 249-256. Trento, Italy.

Doddington, G. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *HLT 2002 (Second Conference on Human Language Technology)*, 128-132. San Diego, CA.

Efron, B., and G. Gong 1983. A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation. *The American Statistician* 37(1): 36-48.

Efron, B., and R. Tibshirani 1993. *An Introduction to the Bootstrap*: Chapman and Hall.

Estrella, P., O. Hamon, and A. Popescu-Belis 2007. How Much Data is Needed for Reliable MT Evaluation? Using Bootstrapping to Study Human and Automatic Metrics In *MT Summit XI*, To appear. Copenhagen, Denmark.

Hamon, O., A. Popescu-Belis, K. Choukri, M. Dabbadie, A. Hartley, W. Mustafa El Hadi, M. Rajman, and I. Timimi 2006. CESTA: First Conclusions of the Technolangue MT Evaluation Campaign. In *LREC 2006 (5th International Conference on Language Resources and Evaluation)*, 179-184. Genova, Italy.

Koehn, P. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *EMNLP '04 (Conference on Empirical Methods in Natural Language Processing)*, 388-395. Barcelona, Spain.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *10th Machine Translation Summit - (MT SUMMIT)*, 79--86. Phuket, Thailand.

Kulesza, A., and S. Shieber 2004. A learning approach to improving sentence-level MT evaluation. In *10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, 75--84. Baltimore MD.

Kumar, S., and W. Byrne 2004. Minimum Bayes-risk decoding for statistical machine translation. In *HLT-NAACL*, 169-176.

Niessen, S., F. J. Och, G. Leusch, and H. Ney 2000. An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *LREC 2000 (2nd International Conference on Language Resources and Evaluation)*, 39-45. Athens, Greece.

Papineni, K. 2002. Machine Translation Evaluation: N-grams to the Rescue. In *LREC 2002 (Third International Conference on Language Resources and Evaluation)*. Las Palmas, Canary Islands, Spain.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. Yorktown Heights, NY: IBM Research Division, T.J.Watson Research Center.

Russo-Lassner, G., J. Lin, and P. Resnik 2005. A Paraphrase-Based Approach to Machine Translation Evaluation. University of Maryland, College Park

Tillmann, C., S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf 1997. Accelerated DP Based Search for Statistical Translation. In *Eurospeech 1997*, 2667--2670. Rhodes, Greece.

Turian, J. P., L. Shen, and I. D. Melamed 2003. Evaluation of Machine Translation and its Evaluation. In *Machine Translation Summit IX*, 386-393. New Orleans, Louisiana, USA.

White, J. S. 2001. Predicting Intelligibility from Fidelity in MT Evaluation. In *Workshop on MT Evaluation "Who did what to whom?" at Mt Summit VIII*. Santiago de Compostela, Spain.

Zhang, Y., S. Vogel, and A. Waibel 2004. Interpreting BLEU/NIST Scores: How Much Improvement do We Need to Have a Better System? In *LREC 2004 (4th International Conference on Language Resources and Evaluation)*, 2051-2054. Lisbon, Portugal.

EBMT Based on Finite Automata State Transfer Generation

Ren Feiliang, Zhang Li, Hu Minghan, Yao Tianshun
NLP Laboratory, Institute of Computer Software and Theory,
Northeastern University, Shenyang, 110004, China
renfeiliang@ise.neu.edu.cn

Abstract

This paper proposes an EBMT method based on finite automata state transfer generation. In this method, first some links from the fragments in the input sentence to the fragments in the target sentence of the selected example are built. Then some predefined states are assigned to these links according to their link types. Finally, taking these links and their corresponding states as inputs, a finite automaton is constructed and the translation result is generated in a finite automata state transfer manner. This method can be easily replicated, and does not need too much complicated parsers either. Based on this method, we built a Chinese-Japanese bidirectional EBMT system to evaluate the proposed method, and experimental results indicate that the proposed method is effective.

1 Introduction

Example-based machine translation (EBMT) is a method of translation by the principle of analogy. It generally consists of three modules: a matching module, an alignment module and a recombination module. Given an input sentence, an EBMT system first matches the input sentence against the example set to select some relevant examples whose source sentence parts are similar to the given input sentence; once the relevant examples have been selected, the alignment module will select the corresponding fragments in the target sentences of the selected examples for every part of the input sentence; once the appropriate fragments have been selected, the recombination module will combine them to form a legal target text (Somers, 1999).

Generally, we can regard the last two modules as a translation generation module. For the generation, some researchers (Aramaki and Kurohashi, 2003; Aramaki and Kurohashi, 2004) used a semantic-based generation approach that obtains an appropriate translation fragment for each part of the input sentence. The final translation is generated by recombining the translation fragments in some order. This approach does not take into account the fluency between the translation fragments. The statistical approach (Akiba et al., 2002; Watanabe and Sumita, 2003; Imamura et al., 2004) selects translation fragments with a statistical model. The statistical model can improve the fluency between the translation fragments by using *n*-gram co-occurrence statistics. However, the statistical model does not take into account the semantic relation between the example and the input sentence. Tree parsing based generation approach (Zhanyi et al., 2005) solves the above two problems by using a method based on tree string correspondence (TSC) and statistical generation. During the translation process of this method, the input sentence is first parsed into a tree. Then the TSC forest is searched to find out if it is best matched with the parse tree. Finally, it uses a statistical generation model to generate translation by combining the target language strings in the TSCs. This method depends heavily on the tree parsing technology, if the parser does not work well, it is impossible to generate a proper translation result.

This paper proposes a generation method for EBMT based on finite automata state transfer. It uses the target sentence of the selected example to generate the translation result in a finite auto-

mata state transfer manner, and outputs the modified target sentence as final translation result.

The rest of this paper is organized as follows. Section 2 gives a brief description of our Chinese-Japanese bidirectional EBMT system. Section 3 describes our generation method in detail. Section 4 presents our experiments. At last, we conclude this paper and present future work in section 5.

2 System Structure of Our Chinese-Japanese Bidirectional EBMT System

Our Chinese-Japanese bidirectional EBMT system's structure is shown in figure 1. A word-based matching method is used to select one example that is most similar to the input sentence. Here two sentences' similarity is calculated as shown in formula 1 (LV Xue-qiang and Ren Feiliang, 2003).

$$Sim(s_1, s_2) = \frac{2 \times SameWord(s_1, s_2)}{Len(s_1) + Len(s_2)} \qquad (1)$$

In this formula, $Sim(s_1, s_2)$ means the similarity of sentence s_1 and sentence s_2, $SameWord(s_1, s_2)$ means the number of common words in sentence s_1 and sentence s_2, and $Len(s_i)$ is the number of total words in sentence s_i.

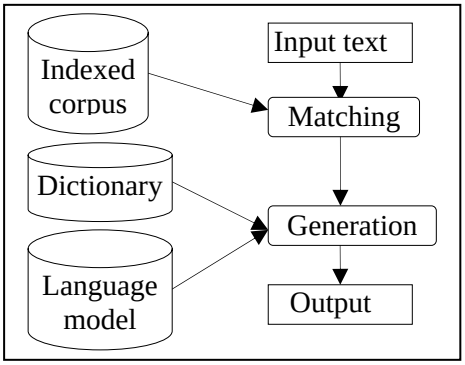


Figure 1. Structure of CJ EBMT System

3 Generation Based on Finite Automata State Transfer

We generate the input sentence's translation by modifying the target sentence of the selected example. This process consists of three steps.

(1) Build links from the fragments in the input sentence to the fragments in the target sentence of the selected example.
(2) Assign states to each of these links.
(3) Construct a finite automaton and generate the translation result in a automaton state transfer manner.

3.1 Building Links

A link from a fragment in one sentence S_1 to a fragment in another sentence S_2 is defined as a 3-tuple (Sf_i, Tf_j, t), where Sf_i (a fragment in S_1), Tf_j (a fragment in S_2), and t are called source fragment, target fragment, and link type respectively. In this 3-tuple, if the languages of S_1 and S_2 are the same, the target fragment is the most similar part in S_2 to the source fragment; if the languages of S_1 and S_2 are different, the target fragment is the most useful part in S_2 to generate the source fragment's translation. Either the source fragment or the target fragment can be null, but they can't be null at the same time. Link type indicates a possible operation converting the source fragment to the target fragment. Following edit distance's style (Wagner and Fischer, 1974), we define four link types: *I, R, D, N*, which mean *inserting, replacing, deleting* and *outputting directly* respectively.

Suppose *S* is an input sentence, *(A, B)* is the selected example. The process of building links from *S's* fragments to *B's* fragments consists of two steps.

(1) Build links from *S's* fragments to *A's* fragments using a revised edit distance algorithm as shown in figure 2. Its result is denoted as *LinkSet(S→A)*.
(2) Build links from *S's* fragments to *B's* fragments (denoted as *LinkSet(S→B)*) according to following rules. (a) For a link in *LinkSet(S→A)*, if neither its source fragment nor its target fragment is null, replace its target fragment with this target fragment's corresponding aligned fragment in *B*, and add this new link to *LinkSet(S→B)*. (b) For a link in *LinkSet(S→A)* whose target fragment is null, add it to *LinkSet(S→B)* directly. (c) For those fragments in *B* that have not been linked, build links for each of them by assigning a null source fragment and a *D* link type to them respectively, and add these links to *LinkSet(S→B)*. (d) Reorder the items of *LinkSet(S→B)* in their target fragments' order in sentence *B*.

In the revised edit distance algorithm, it takes fragments as comparison units, and its two input sentences *S* and *A* are segmented into fragments by two segmentation tools[1] before they are inputted. This is a little different from Brown (1996) who took a full segmentation strategy for the input sentence.

```
m=length(S1), n=length(S2)
d[0][0] =0; tags[0][0] = 0;
for i=1 to m
  d[i][0]=q+d[i-1][0]; tags[i][0]='D'
for j=1 to n
  d[0][j]=r+d[0][j-1]; tags[0][j]='I'
for i=1 to m
 for j=1 to n
   p = computeCost(S1[i-1],S2[j-1]);
   a = d[i-1][j-1] + p;
   b=d[i-1][j] + q;
   c=d[i][j-1] + r;
   d[i][j] = min(a,b,c);
   if(min==a and p==0)
      tags[i][j] = 'N';
   else if (min==a)
      tags[i][j] = 'R';
   else if (min==b)
      tags[i][j] = 'D';
   else if (min==c)
      tags[i][j] = 'I';
return tags
```

Figure 2. Revised Edit Distance Algorithm

In figure 2, ***computeCost*** is a function to compute two fragments' linking cost based on their lexical forms and their head words' POSs. Its possible value belongs to the range [0, 1] and is manually assigned according to human's experiences. If two fragments' lexical forms are the same and their head words' POSs are the same too, this cost is zero; if two fragments' lexical forms are the same but their head words' POSs are different, this cost is 0.2; otherwise, this value is assigned by human's experiences according to the two fragments' head words' POSs as shown in table 1.

Table 1. Linking Cost for Two Fragments

$PosPair(c_i, c_j)$	w_i
(noun, noun)	0.5
(noun, auxiliary)	0.8
(noun, adjective)	0.85
…	…

[1] http://chasen.aist-nara.ac.jp/hiki/ChaSen/ for Japanese
http://www.nlplab.com/chinese/source.htm for Chinese

In figure 2, q, r are constants. It is required that $q + r \geq p$ and $q,r \in (0,1]$, here we set $q = r = 1$. The returned ***tags*** is *LinkSet(S→A)*.

After step 1, we can build links from sentence S to sentence B according to the rules described in step 2, and an example of this process is shown in figure 3.

Suppose *S* is "他很爱他的妻子*(He loves his wife very much)*". The selected example *(A,B)* is "(他爱他的妈妈*(He loves his mother)*, 彼は、彼の母を愛しています*(He loves his mother)*)".

Firstly, *LinkSet(S→A)* is built using the algorithm shown in figure 2. It is: (他*(he)*,他,*N*), (很*(very much)*,null,*I*), (爱*(loves)*,爱,*N*), (他的*(his)*,他的, *N*), (妻子*(wife)*,妈妈*(mother)*,*R*).

Secondly, *LinkSet(S→B)* is built as follows. We know that in *(A,B)*, "他"aligns to "彼*(he)*", "爱" aligns to "愛しています*(loves)*" , "他的" aligns to "彼の*(his)*", and "妈妈" aligns to "母*(mother)*", according to rule (a), we replace these target fragments in *LinkSet(S→A)* with their corresponding aligned fragments in *B* and add them to *LinkSet(S→B)*, and *LinkSet(S→B)* is changed to: (他*(he)*, 彼*(he)*,*N*), (爱*(loves)*, 愛しています*(loves)*,*N*), (他的*(his)*,彼の*(his)*,*N*), (妻子*(wife)*,母*(mother)*,*R*). For the link (很*(very much)*,null,*I*), according to rule (b), we add it to *LinkSet(S→B)* directly. Besides, there are some fragments in *B* that haven't been linked, according to rule (c), we build links for each of them by assigning them a null source fragment and a link type *D*, and add these new links in *LinkSet(S→B)* , and *LinkSet(S→B)* is changed to: (他*(he)*,彼*(he)*,*N*), (爱*(loves)*,愛しています*(loves)*,*N*), (他的*(his)*, 彼の*(his)*, *N*), (妻子(wife), 母*(mother)*,*R*), (很*(very much)*,null,*I*), (null,は*(ha)*,*D*), (null,を*(wo)*,*D*). At last, according to rule (d), we reorder the items in *LinkSet(S→B)*, and the final *LinkSet(S→B)* is: (他*(he)*, 彼*(he)*,*N*), (null, は*(ha)*,*D*), (很*(very much)* ,null,*I*), (他的*(his)*, 彼の*(his)*, *N*), (妻子(wife), 母*(mother)*,*R*), (null,を*(wo)*,*D*), (爱*(loves)*,愛しています*(loves)*,*N*).

Figure 3. An Example of Building Links

3.2 States Assignment

3.2.1 States for Non-*I* Type's Links

If a link's type is not *I*, that is to say it is one of the types {*R, D, N*}, the state assignment is easy. If its link type is *R*, a state named *S_R* is assigned;

if its link type is *D*, a state named *S_D* is assigned; if its link type is *N*, a state named *S_N* is assigned.

3.2.2 States for *I*-Type's Link

For an *I*-type's link, it indicates a possible generation operation is inserting. Different from other link types, there are two challenges for it: one is how to select a proper inserting position; the other is how to make the whole sentence fluent when finishing this inserting operation. In response to these two problems, we use current *I*-type link's pre- and post- links' link shapes to define current *I*-type link's state.

Suppose an *I*-type's link in *LinkSet(S→B)* is (*i*, null,*I*), *i*+1 and *i*-1 are the post- and pre- fragments of this link's source fragment. *m* and *n* are some fragments in sentence *B*. It is the same that we use $m \pm 1$ and $n \pm 1$ to denote the post- and pre- fragments of m and n respectively.

According to the link shapes of the links that take *i*+1 and *i*-1 as their source fragments, there are twelve basic link shapes shown in figure 4 and three extended link shapes shown in figure 5. We map each of these link shapes to an *I*-type link's state. Thus there are twelve basic states and three extended states for *I*-type's links.

In figure 4 and figure 5, a dot rectangle denotes a true link in *LinkSet(S→B)*, and a bold rectangle denotes this link's generation path when taking into account *LinkSet(S→A)*.

A brief explanation to these states is as follows. For example, state 6 in figure 4 means *S's* fragment *i*-1 links to *B's* fragment *m* and *S's* fragment *i+1* links to nothing in *B*. The appearance reason for this null target fragment is that in sentence pair *(S,A)*, fragment *i+1* links to fragment b_j, but in sentence pair *(A,B)*, b_j aligns to null, thus *i+1* links to null according to the second step when building *LinkSet(S→B)*. Due to the same or similar reason, state 7, 8, 10, 12,13,14,15 also have null target fragments in their links. We distinguish these link shapes because they will be treated differently. State 9 indicates that *i* is the first fragment in sentence *S*. State 11 indicates that *i* is the last fragment in sentence *S*.

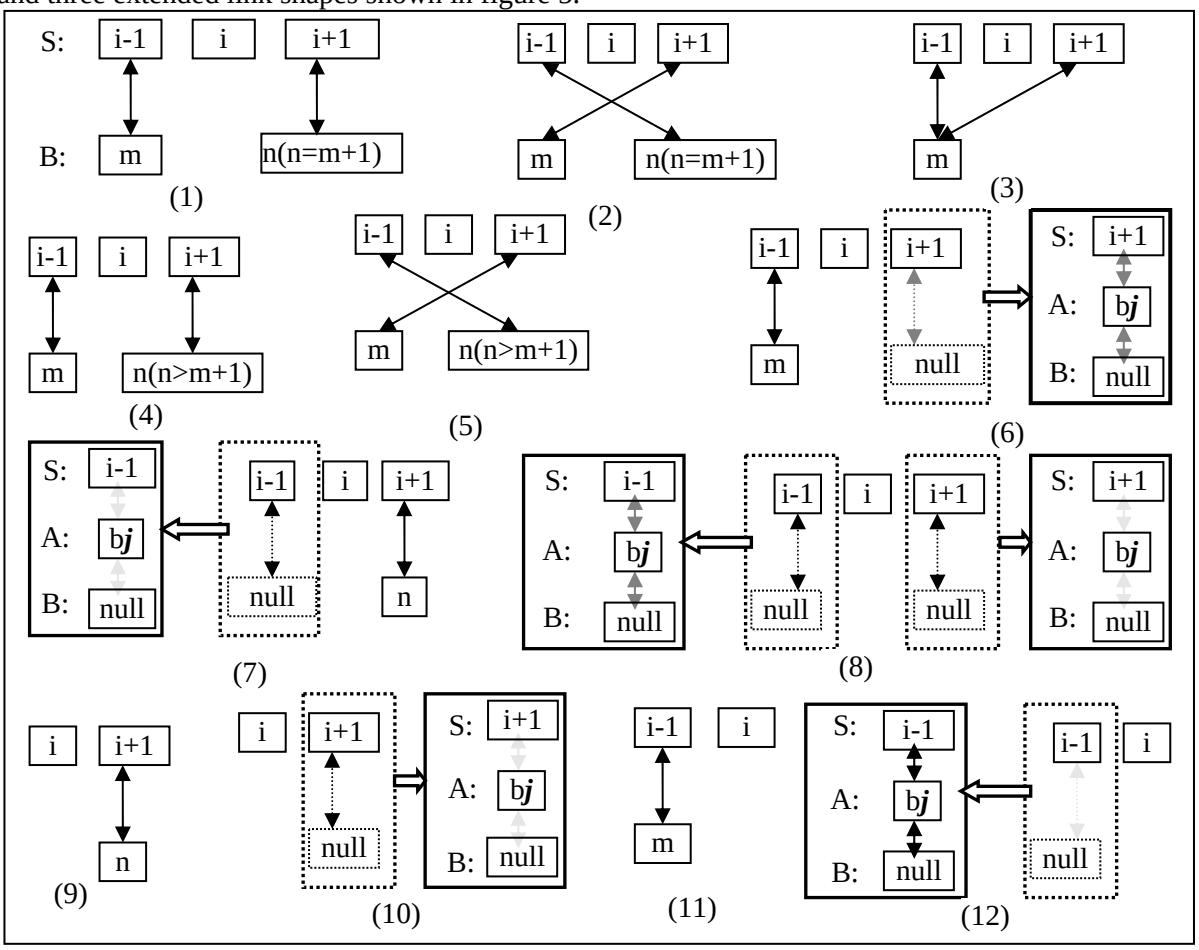


Figure 4. Basic States for *I*-type's Link

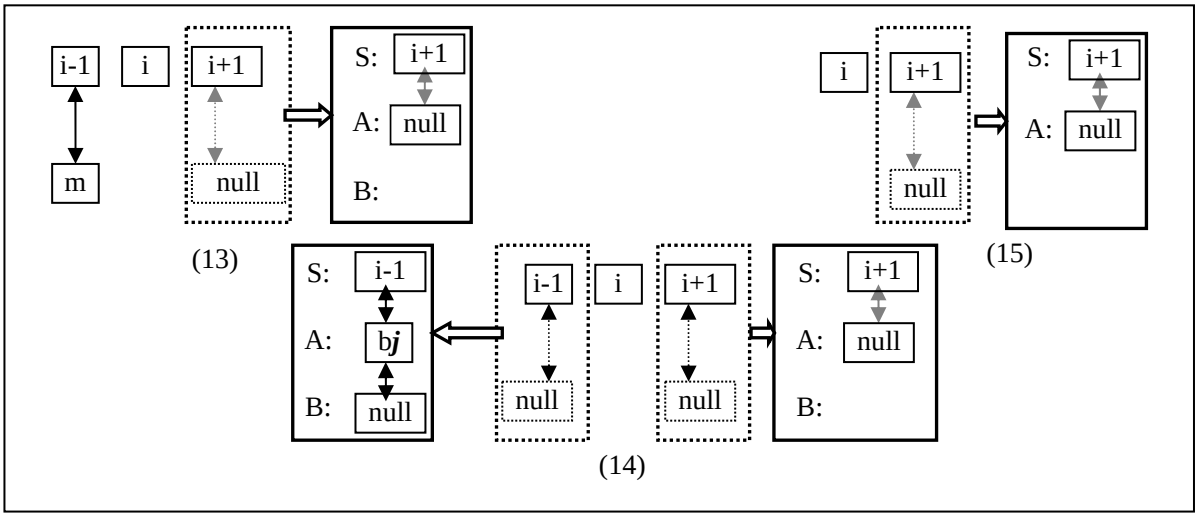


Figure 5. Extended States for *I*-type's Link

In practice, we will meet the extended states in figure 5, but they can be converted into basic states in some way. These conversion rules are as follows. For state 13, move rightward until find a non-*I* type's link, if this link's target fragment is null, convert it to state 6; otherwise, convert it to a state among state 1 to state 5 according to the link shapes of fragment *i*-1's link and the new found link; if can't find a non-*I* type's link in current link's right side, convert it to state 11. For state 14, move rightward until find a non-*I* type's link, if this link's target fragment is null, convert it to state 8, otherwise, convert it to state 7; if can't find a non-*I* type's link in current link's right side, convert it to state 12. For state 15, move rightward until find a non-*I* type's link, if this link's target fragment is null, convert it to state 10, otherwise, convert it to state 9; if can't find a non-*I* type's link in current link's right side, move leftward until find a non-*I* type's link (this link will be found always) and convert it to state 11.

For all these conversions, the final new state's *I*-type link takes all the passed fragments in *S* during rightward movement as its new source fragment.

By conversion, every *I*-type's link can be mapped to a basic state in figure 4, and we can consider basic states only in the following description.

3.3 Translation Generation

In this process, an automaton is constructed to generate the input sentence's translation. For different state, there is different generation operation corresponds to it.

3.3.1 Generation Operations for Non-*I* Type Links' States

If a link's type is not *I*, we take an easy generation strategy according to its state. If a link's state is *S_R*, replace this link's target fragment with its source fragment's translation, and denote this operation as ***O(R)***; if a link's state is *S_D*, delete this link's target fragment, and denotes this operation as ***O(D)***; if a link's state is *S_N*, remain this link's target fragment unchanged, and denote this operation as ***O(N)***. Here a link's source fragment's translation is generated by looking up a dictionary.

3.3.2 Generation Operations for *I*-Type Links' States

If a link's type is *I* (suppose its source fragment is *i*), we take its source fragment's pre- and post-fragments into account and judge: whether the fragment combinations *(i-1,i,i+1)*, *(i-1,i)* and *(i,i+1)* are chunks. If they are chunks, look up their corresponding translations in dictionary, otherwise, look up *i's* translation in dictionary (we assume its translation can be found always). Here a chunk is defined as a translation unit and a simple dictionary-based method is used for chunk recognition: as long as a fragment can be found in dictionary, it is regarded as a chunk. According to current *I*-type link's state and the recognized chunk information, we choose one of these chunks as current *I*-type link's new source fragment for later processing, and define 10 possible generation operations as follows.

- ***O(0):*** Delete the links that take *B's* fragments among *m+1* to *n* as their target

fragments. And for the link that takes *B's* fragment *m* as target fragment, replace *m* with the translation of current *I*-type link's new source fragment.

- ***O(1***): For the link that takes *B's* fragment *m* as target fragment, replace *m* with the translation of current *I*-type link's new source fragment.
- ***O(2):*** For the link that takes *B's* fragment *n* as target fragment, replace *n* with the translation of current *I*-type link's new source fragment.
- ***O(3):*** For the link that takes *B's* fragment *m* as target fragment, add the translation of current *I*-type link's new source fragment to the end of *m*.
- ***O(4):*** For the link that takes *B's* fragment *n* as target fragment, add the translation of current *I*-type link's new source fragment to the end of *n*.
- ***O(5):*** For the link that takes *B's* fragment *m* as target fragment, replace *m* with the translation of current *I*-type link's new source fragment. And delete the link that takes *B's* fragment *n* as target fragment.
- ***O(6):*** For the link that takes *B's* fragment *n* as target fragment, replace *n* with the translation of current *I*-type link's new source fragment. And delete the link that takes *B's* fragment *m* as target fragment.
- ***O(7):*** For the link that takes *B's* fragment *m* as target fragment, add the translation of current *I*-type link's new source fragment before *m*.
- ***O(8):*** For the link that takes *B's* fragment *n* as target fragment, add the translation of current *I*-type link's new source fragment before *n*.
- ***O(9):*** Do not modify any link's target fragment.

Here *m* and *n* are sentence *B's* fragments, and they also correspond to the target fragments of the links shown in figure 4.

During the generation, which operation should be chosen depends on current *I*-type link's state and the result of chunk recognition. The choice strategy will be described subsequently.

3.3.3 Finite Automaton State Transfer Based Generation

Based on *LinkSet(S→B)* and the assigned states, we construct an automaton that has a similar form as shown in figure 6. This automaton takes *LinkSet(S→B)* and the assigned states as input, executes generation operations according to these states and outputs *LinkSet(S→B)'s* final modified target fragment sequence as the input sentence's translation result.

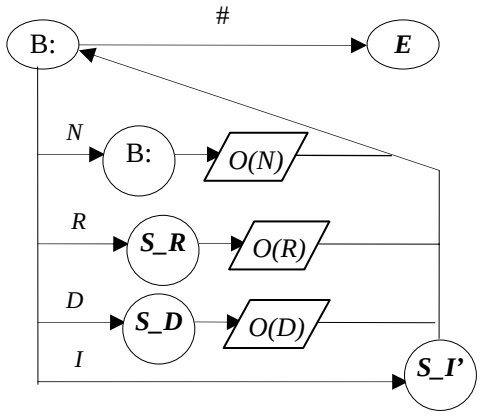


Figure 6. Finite Automation State Transfer Based Generation

In figure 6, ***B*** is a start state, ***E*** is an end state, {*I, R, D, N*} are link types, {*O(N), O(D), O(R)*} in parallelogram are the operations defined in section 3.2.1; and # is a fictitious symbol that indicates the end of the automaton's input. {*S_R, S_D, S_N*} are states correspond to non-*I* type's links. And *S_I'* is a state set that corresponds to *I*-type's links. When the state transfers to *S_I'*, the corresponding operations are shown in figure 6. In this figure, numbers from *1* to *12* in ellipse circles correspond to the states defined in figure 4. *O(i)* in parallelogram corresponds to the operations defined in section 3.3.2; *O'* in the operation of state 3 means the automaton generates the fragment combination *(i-1,i,i+1)'s* translation by simply joining their single fragment's translations together. d_1 means the semantic distance from fragment *i* to fragment *i-1*, and d_2 means the semantic distance from fragment *i* to fragment *i+1*, and they are computed as shown in formula 2.

$$dist(f_1, f_2) = \sum_{c_i \in f_1} \sum_{c_j \in f_2} w_k(PosPair(c_i, c_j)) \quad (2)$$

In formula 2, f_1 and f_2 are fragments, c_i and c_j are words in them, w_k is a weight function whose value is determined by the POSs of words c_i and c_j, and its value assignment strategy can be referred to table 1. When current *I*-type link's pre- and post- links' target fragments span several fragments, this formula is used to identify a proper inserting position for the translation of current *I*-type link's source fragment. The larger this distance, the less possibility its two fragments' translations are close.

Figure 7 shows the operation strategy for different states of the *I*-type's links. Here we take state 1 as example and give some explanations for these operations in figure 6. For state 1, if the fragment combination *(i-1,i,i+1)* is a chunk, from the link shape of state 1 in figure 4 we can see, there is a strong hint that the original target fragments of the two links that take fragments *i-1* and *i+1* as their source fragments respectively should be replaced by this new chunk's translation, and this just corresponds to the first operation defined in section 3.3.2. Otherwise, if *(i-1,i)* is a chunk, there is a strong hint that the original target fragment of the link that takes *i-1* as its source fragment should be replaced by this new chunk's translation; and other cases are similar to these explanations.

The main idea for the operation strategies in figure 7 is trying to enlarge the source fragment for an *I*-type's link, and using its contextual links' link shapes to find a proper inserting position for the translation of its new source fragment.

To demonstrate this generation process, we continue the example introduced in section 3.1.

After building links described in section 3.1 *LinkSet(S→B)* is: (他*(he)*, 彼*(he)*,*N*), (null, は*(ha)*,*D*), (很*(very much)*,null,*I*), (他的*(his)*, 彼の*(his)*,*N*), (妻子(wife),母*(mother)*, *R*), (null, を*(wo)*,*D*) (爱*(loves)*,愛しています*(loves)*,*N*).

Its corresponding state sequence is: *S_N, S_D, S_I_4(*the forth state in figure 4*),S_N, S_R, S_D, S_N*.

During the process of generation, the constructed automaton takes *LinkSet(S→B)* and the corresponding state sequence for the links in *LinkSet(S→B)* as inputs, and analyzes these inputs one by one. This process is shown in figure 8 which give an example of the translation generation process.

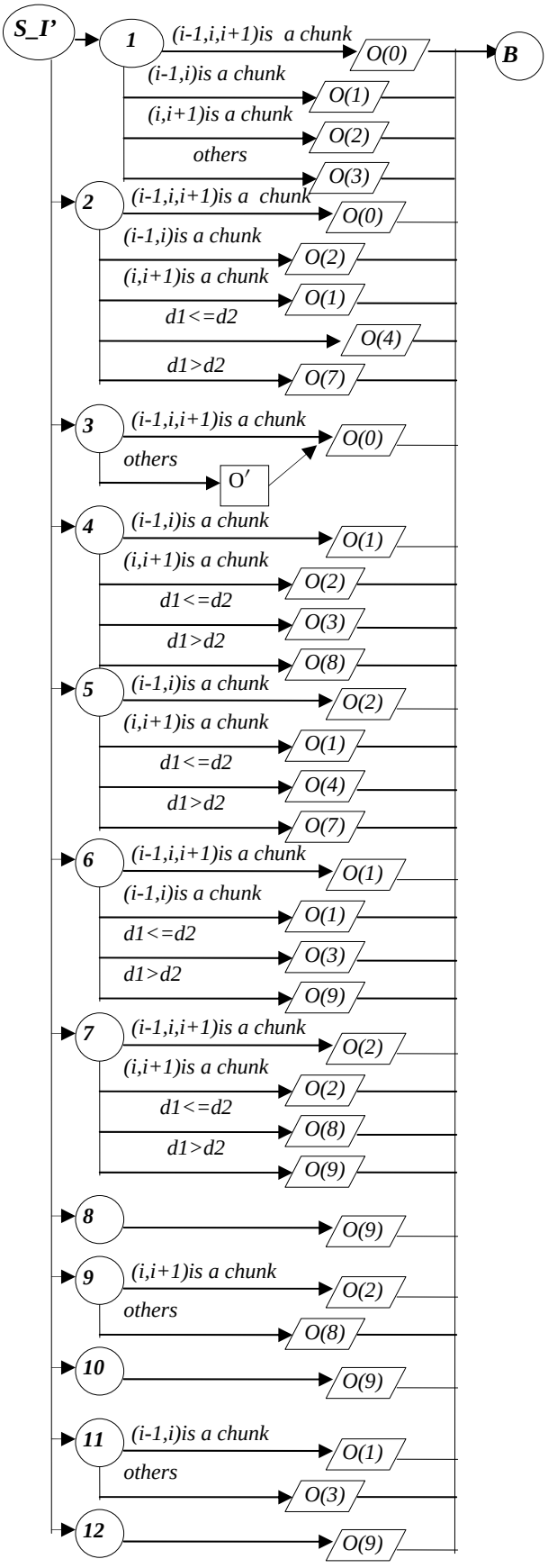


Figure 7. State Transfer for *I*-Type's Links

For the link (他*(he)*,彼*(he)*,*N*), its state is *S_N*. According to figure 6, the automaton executes operation *O(N)* and does not modify this link's target fragment.

For the link (null,は*(ha)*,*D*), its state is *S_D*. According to figure 6, the automaton executes operation *O(D)* and deletes this link's target fragment.

For the link (很*(very much)*,null,*I*), its state is *S_I_4*. If the fragment combination *(i-1,i)* "他很*(he…very much)*" is a chunk and the corresponding translation is "彼は、とても *(he…very much)* ", according to figure 6, the automaton executes operation *O(1)*. It first takes this recognized chunk as current link's new source fragment. Then it selects the link whose target fragment is "彼*(he)*", and this link is (他*(he)*,彼*(he)*,*N*). Thirdly, it replaces the selected link's target fragment with the translation of current *I*-type link's new source fragment. At last the selected link is changed to (他*(he)*, 彼は、とても*(he…very much)*, *N*).

For the link (他的*(his)*,彼の母*(his)*,*N*), its state is *S_N*. According to figure 6, the automaton executes operation *O(N)* and does not modify this link's target fragment.

For the link (妻子*(wife)*,母*(mother)*,*R*), its state is *S_R*. According to figure 6, the automaton executes operation *O(R)* and replaces this link's target fragment with its source fragment's translation. Finally current link is changed to (妻子*(wife)*,妻*(wife)*,*R*).

For the link (null,を*(wo)*,*D*), its state is *S_D*. According to figure 6, the automaton executes operation *O(D)* and deletes this link's target fragment.

For the link (愛*(loves)*,愛しています*(loves)*,*N*), its state is *S_N*. According to figure 6, the automaton executes operation *O(N)* and does not modify this link's target fragment.

At last, the automaton ends the state transfer process and outputs *LinkSet(S→B)'s* modified target fragment sequence "彼は、とても彼の妻愛しています*(he loves his wife very much)*" and takes it as the input sentence's translation.

Figure 8. An Example of Generation

4 Experiments

We developed a Chinese-Japanese bidirectional EBMT system to evaluate the proposed method in term of translation quality, and BLEU value and NIST score are used for evaluation. The evaluation tool is the NIST MT Evaluation Toolkit[2].

4.1 System Resources

Bilingual Corpus We collect 10083 Chinese-Japanese bilingual sentences from Internet in Olympic domain as examples. The average length of the Chinese sentences is 12.8 characters while the average length of the Japanese sentences is 25.6 characters. All the examples are stored in their lexical form along with their fragments alignment information. We used an in-house tool for fragment alignment and revised this result by some experienced experts.

Bilingual Dictionary A bilingual dictionary is used to translate the input fragment and to judge whether an input fragment is a chunk.

This bilingual dictionary contains not only the general word items, but also some bilingual chunks collected from our corpus by an in-house rule-based chunk parser. All together there are about 150,000 word items and about 71,000 chunk items in this bilingual dictionary.

Language Model During the process of *R*-type and *I*-type links' generations, if a fragment has several translations, a language model is used for its translation choice (Feiliang Ren and Tianshun Yao, 2006). Its work principle is to make the whole sentence fluent most after fragments translation choices. For example, if during the process of translation generation, we need to insert a fragment's translation into the target part of the selected translation example, and if there are several different translations for this fragment in dictionary, which translation should be chosen? Our method is to choose the one that can make the final sentence fluent most after choices. And use language model to measure the fluency of a sentence.

We collected an approximate 1,400,000 words' Japanese monolingual corpus and a similar size's Chinese monolingual corpus from Internet, and trained a standard trigram Japanese language model for Chinese-to-Japanese EBMT system and a standard trigram Chinese language model for Japanese-to-Chinese EBMT system respectively.

Test Corpus We collect another 100 bilingual sentences in Olympic domain from Internet as

[2] ttp://www.nist.gov/speech/tests/mt/resources/scoring.htm

test corpus. But it is required that for every sentence *S* in test corpus, there must be at least one example *(A, B)* that satisfies $Sim(S, A) \geq 0.4$. This is because the characteristic of EBMT is translated by analogy. If there weren't any proper examples for the input sentence, the advantage of EBMT would vanish. When this happened, system would have to perform translation in a different manner. This is not what we hope. This threshold condition can guarantee that system performs translation in an EBMT manner and thus we can focus on the generation method proposed.

4.2 Experimental Results

We take system's matching module as a baseline system. In fact it is a TM (translation memory) system, and its performance is the lowest limit of our translation system's performance.

In the evaluation, we set N at 4 when computing BLEU value and NIST score. Experimental results for Chinese-to-Japanese EBMT system and Japanese-to-Chinese EBMT system are shown in table 2 and table 3 respectively.

Table 2. Experimental Results for Chinese-to-Japanese EBMT System

Method	NIST	BLEU
Baseline	4.8321	0.4913
Our System	***5.9729***	***0.7705***

Table 3. Experimental Results for Japanese-to-Chinese EBMT System

Method	NIST	BLEU
Baseline	4.1275	0.4076
Our System	***5.0976***	***0.5908***

From table 2 and table 3, it can be seen that our system achieves excellent translation performances in both Chinese-to-Japanese translation system and Japanese-to-Chinese translation system. These results are unexpected and encouraging. We think the following reasons lead to these good results. First, we set a threshold in matching module. This guarantees that even under the worst condition, our system's performance is still at a relative high level. Second, the alignment results for the fragments of the examples stored in corpus are revised by experienced experts. It makes the alignment precision be very high. And this is very helpful when building links before generation. Third, we generate the translation by modifying the target sentence of the selected example, this makes us use the existed target sentence's structure information as much as possible, and it is useful for generating translation that conforms to the grammar and the semantic rules well. Forth, the most important point is that we view the generation as a process of finite automata state transfer, search out the most useful information for the input fragments by building fragments' links from the input sentence to the target sentence of the selected example, and take different generation strategies for different kinds of states.

We also notice that the performance of Chinese-to-Japanese translation system is better than the performance of Japanese-to-Chinese translation system. This is because that generally a Japanese sentence has a more complicated structure than a Chinese sentence. This will lead to poorer result when building fragments' links from sentence *S* to sentence *A*, thus the fragments' links from *S* to *B* are worse accordingly. So the final translation result will be worse because the proposed method is affected by the link result heavily. More work should be done to improve the algorithm that builds links from *S's* fragments to *A's* fragments.

Besides, there are still some translation results that are not as good as expected. For example, in the Chinese-to-Japanese translation system, some auxiliary particles were wrongly deleted, which made several translation results were somewhat odd when checked by a Japanese native speaker. This is caused by the simple deleting strategy in our generation process for those *D*-type's links. We think that operation strategy for these *D*-type's links needs further improvement.

5 Conclusions and Future Work

This paper proposes an EBMT method based on finite automata state transfer generation. During the translation process, first a bilingual sentence pair is selected as example whose source sentence is most similar to the input sentence; then the target sentence of this example is used to generate final translation result in a finite automata state transfer manner. During the generation process, firstly we build links from the fragments in the input sentence to the fragments in the target sentence of the selected example. Then we assign states for each of these links. Finally, we construct a finite automaton with these states and generate a translation result in a finite automata state transfer manner. Our method hasn't any special requirement for corpus's domain. It can

be easily replicated, and does not need some complicated parsers either. As long as you have a bilingual corpus and a fragment alignment technology (even it is a simple dictionary-based method), you can replicate our work. Therefore, we think it is a good baseline method for machine translation.

From the generation process and experimental results we can see that there are some factors that affect our translation system's performance heavily, such as the algorithm used to build links, the similarity algorithm for matching module, the fragment alignment technology, and the chunk recognition method and the translation generation technology for the recognized chunks, and so on. In future work, we will investigate improving the performances of these factors.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No.60473140; the National 863 High-tech Project No.2006AA01Z154; the Program for New Century Excellent Talents in University No.NCET-05-0287; and the National 985 Project No.985-2-DB-C03.

References

Eiji Aramaki, Sadao Kurohashi. 2003. Word Selection for EBMT based on Monolingual Similarity and Translation Confidence. In Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts, pp 57-64

Eiji Aramaki, Sadao Kurohashi. 2004. Example-Based Machine Translation Using Structural Examples. International Workshop on Spoken Language Translation (IWSLT), pp. 91-94.

Eric Sven Ristad, Peter N.Yianilos. 1998. Learning string-edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence. 20(5):522-532

Feiliang Ren, Tianshun Yao. 2006. Make Word Sense Disambiguation in EBMT Practical. The 20th Pacific Asia Conference on Language, Information and Computation. pp414-417

Harold Somers.1999. Review Article: Example-based Machine Translation. Machine Translation 14, pp.113-157

Kenji Imamura, Hideo Okuma, Taro Watanabe and Eiichiro Sumita. 2004. Example-Based Machine Translation Based on Syntactic Transducer with Statistical Models. In Proc. of the 20th International Conference on Computational Linguistics (COLING-2004), pp. 99-105.

Kevin Knight, Yaser Al-Onaizan, 1998. Translation with Finite-State Devices, Proceedings of the AMTA Conference, Langhorne, PA, USA. pp.421-437.

Liu Zhanyi, Wang Haifeng, Wu Hua. 2005. Example-based machine translation based on TSC and statistical generation. MT Summit X, pp.25-32

LV Xue-qiang, Ren Feiliang, 2003. Sentence Similarity Model and the Most Similar Sentence Search Algorithm. Journal of Northeastern University (Natural Science). Pp531-534

Ralf D. Brown, 1996. Example-Based Machine Translation in the Pangloss System, Procddeings of the 16th International Conference on Computational Linguistics(COLING-96), pp.169-174

Robert A.Wagner, Michael J.Fischer. 1974. The String-to-String Correction Problem, Journal of the ACM(JACM), v.21, pp.168-173

Shankar Kumar, William Byrne. 2003. A Weighted Finite state automata transfer Implementation of the Alignment Template Model for Statistical Machine Translation. Proceedings of the Conference on Human Language Technology. Edmonton, Canada. Pp.142-149.

Shankar Kumar, William Byrne, 2004. A Wighted Finite state automata transfer Example Model for Statistical Machine Translation. Natural Language Engineering

Srinivas Bangalore, Giuseppe Riccardi. 2001. A finite state approach to machine translation. Proceedings of the 2nd meeting of North American Chapter of the Association for Computational Linguistics. Pittsburgh, PA, USA.

Taro Watanabe, Ei ichiro Sumita. 2003. Example-based decoding for statistical machine translation. In Proceedings of MT Summit IX, pp.410-417

T.H.Cormen, C.E.Leiserson, R.L. Rivest and C.Stein. 2001. Introduction to Algorithms (the second edition). The MIT Press

Yasuhiro Akiba, Taro Watanabe and Eiichiro Sumita. 2002. Using Language and Translation Models to Select the Best among Outputs from Multiple MT systems. In Proceedings. of the 19th International Conference on Computational Linguistics (COLING-2002), pp. 8-14.

Learning Bilingual Semantic Frames: Shallow Semantic Parsing vs. Semantic Role Projection

Pascale Fung, Zhaojun Wu, Yongsheng Yang, and Dekai Wu
Human Language Technology Center
Hong Kong University of Science and Technology
pascale@ee.ust.hk, {josephwu, ysyang, dekai}@cs.ust.hk

Abstract

To explore the potential application of semantic roles in structural machine translation, we propose to study the automatic learning of English-Chinese bilingual predicate argument structure mapping. We describe ARG_ALIGN, a new model for learning bilingual semantic frames that employs monolingual Chinese and English semantic parsers to learn bilingual semantic role mappings with 72.45% F-score, given an unannotated parallel corpus. We show that, contrary to a common preconception, our ARG_ALIGN model is superior to a semantic role projection model, SYN_ALIGN, which reaches only a 46.63% F-score by assuming semantic parallelism in bilingual sentences. We present experimental data explaining that this is due to cross-lingual mismatches between argument structures in English and Chinese at 17.24% of the time. This suggests that, in any potential application to enhance machine translation with semantic structural mapping, it may be preferable to employ independent automatic semantic parsers on source and target languages, rather than assuming semantic role parallelism.

1 Introduction

As statistical language learning technologies strain the limits of the relatively flat, simplistic structures of first-generation models, the need to incorporate representations that capture meaningful semantic patterns has become increasingly evident. Particularly for cross-lingual applications, techniques for multilingual semantic parsing and the acquisition of cross-lingual semantic frames have numerous potential applications. Error analysis suggests that a structured bilexicon containing a large inventory of cross-lingual semantic frame argument mappings—rather than merely word or phrase translations—would be invaluable toward attacking common types of errors in statistical machine translation, machine-aided translation, or cross-lingual information extraction or summarization models.

For example, inspection of recent contrastive error analysis data from a typical phrase-based SMT system shows that around 20% of the incorrect translations produced could have been avoided if the correct predicate argument information had been used (Och et al., 2003). Consider the following example from the error analysis data:

input 美国政府今天表示，有关美国要求澄清报导以色列意图在所占领的戈兰高地扩大犹太人的屯垦计划，以色列尚未给予满意的回答。

system The United States Government requested clarification of Israel's intention

in the occupied Golan today, on the planned expansion of Jewish settlement, Israel has not yet given a satisfactory response.

reference The United States government said today that Israel had not provided a satisfactory answer to U.S. request for clarification about the reported plans to expand Jewish settlement in the occupied Golan Heights.

This example exhibits a typical mistake arising from the system's lack of awareness of the correct argument structure for the nominalized "*intention*" verb frame (as well as numerous other complements). Such errors of semantic role confusion are one of the most common sources of errors in current statistical systems that rely only on relatively flat representational structures and n-gram language models. Different languages realize semantic roles using different surface forms, and the language models and word reordering models in SMT are not always sufficient to discriminate between alternative hypotheses that may score equally well in fluency despite high variance in translation adequacy.

Bilingual frame semantics, if available, would provide an additional source of translation disambiguation leverage required to attack such problems. This necessitates the cross-lingual acquisition of a large inventory of *bilingual semantic frames*, which capture the needed role correspondence information in a manner independently of word reordering. Bilingual semantic verb frames specify the conventional patterns of alignment of semantic argument structures between a pair of semantic frames (or valency frames, qualia structures, etc.) for verbs in translation.

A challenge we faced is that (contrary to what one might first assume) even with semantic rather than syntactic arguments, the acquisition model still needs to be capable of dealing with the fact that predicate verb translations in English and Chinese often do *not* have the same semantic argument structure, due to cross-linguistic lexical and conceptual differences and translation idiosyncrasies. That is, the *ARG1* (say) in the Chinese semantic verb frame may not align to the *ARG1* in the frame for the corresponding English verb. This might seem surprising since, in principle, it would seem that semantic role labels for translatable verbs ought to be preserved more closely than syntactic roles across languages, since the agents, patients, and so forth seem more likely to remain constant in translation independent of verb alternations—whereas in contrast, surface syntactic labels (subject, object, etc.) often do not survive translation, due to language-specific verb alternations. However, we will describe experimental results indicating that even semantic roles are not preserved across Chinese and English 17.24% of the time.

Thus, our acquisition model cannot assume that the argument labels (*ARG0*, *ARG1*, ...) learned by our separately trained Chinese and English semantic parsers will necessarily correspond to each other cross-linguistically. To address this we introduce a cosine similarity model enabling our acquisition model to build and extract the bilingual semantic verb predicate-argument structure. We then compare this model to a semantic role projection model that uses syntactic constituent alignment, and which preserves semantic roles cross-lingually.

This paper is organized as follows. We begin by defining the bilingual semantic frame mapping problem. In section 3, we describe our findings from a manually aligned reference set of semantic structure mappings. Section 4 presents our new approach to semantic frame mapping, ARG_ALIGN, followed by the experimental results in section 5. In section 6, we then demonstrate experimentally how ARG_ALIGN outperforms a more conventional method based on semantic role projection, SYN_ALIGN.

2 Problem Definition

In recent years, researchers have shown that statistical machine translation models can be enhanced by incorporating structural information (Wu and Chiang, 2007). The atten-

tion, though, has thus far been largely focused on chunk or syntactic structures. Researchers only recently began seriously investigating whether incorporating *semantic* models can enhance statistical machine translation performance (Carpuat and Wu, 2005a; Carpuat and Wu, 2005b), and are only just beginning to show that semantic word sense disambiguation techniques can indeed improve accuracy (Carpuat et al., 2006; Carpuat and Wu, 2007). However, it remains an intriguing open question as to how semantic *structures*—semantic role mappings in bilingual semantic frames—can also be potentially leveraged to improve machine translation.

Thus, in order to overcome the immediate obstacle to exploring this potential, we are interested in learning the bilingual semantic structure given a predicate verb pair in English and Chinese, as in Figure 1. The predicate verb pair "*organized*/举办" have the operators *ARG0* "*African Environmental Centre*/非洲环境中心", and the operands *ARG1* "*Seminar on desertification*/沙漠化问题研讨会".

In the above example, the subject of the English sentence is *ARG1*, the operand, whereas the object is *ARG0*, the operator. On the other hand, the subject-object order is reversed in the Chinese sentence. The location "*Ivory Coast*" after the predicate verb and *ARG1*, at the end of the English sentence, whereas the Chinese translation is before the predicate verb, after *ARG0*, in the Chinese sentence. We are interested in learning and acquiring bilingual semantic frame mapping as illustrated in the above example, as an additional knowledge source for structural machine translation.

3 Findings in the Oracle Semantic Frame Mapping

To facilitate the development and evaluation of bilingual semantic frame acquisition methods, it was necessary for us to create an annotated gold standard reference corpus, containing parallel sentences whose semantic predicates and arguments are not only labeled but also mapped between Chinese and English.

Table 1: Reference Semantic Role Mappings

EN\CN	ARG0	ARG1	ARG2	ARG3
ARG0	326	77	7	1
ARG1	21	540	48	0
ARG2	3	28	39	2
ARG3	0	1	1	1

We aligned the semantic verb frames cross-lingually from a subset of the pre-release version of the Parallel Proposition Bank II for Chinese and English (Palmer et al., 2005). The Parallel Proposition Bank II for Chinese and English is derived from the Chinese Treebank English Parallel Corpus. Both the Chinese sentences and their English translations have been annotated syntactically in the Treebank format and semantically in the PropBank format.

We construct an oracle semantic role mapping based on manual semantic role alignment. The mapping matrix is shown in Table 1. Only the mapping between major core arguments (from *ARG*0 to *ARG*3 in the Proposition bank) are of interest at this stage. This is owing to the fact that, although the Chinese Propbank contains over 40 argument types and the English Propbank over 200, only core arguments *ARG*0 to *ARG*5 are responsible for representing the main semantic concepts, other argument types are served as adjunctive components (referred to as ARGM) that are used to provide additional information, for instance, ARGM-TMP for temporals. According to our observation, the occurrences of these core arguments diminish drastically after number 3.

As we can see from Table 1, around 82.74% of the mappings are direct mapping from ARG_i in English to ARG_i in Chinese. However, there remain a significant proportion of mappings that do not agree with direct mapping. Specifically, around 8.95% of the role mappings are from ARG_0 to ARG_1, 6.94% are from ARG_1 to ARG_2, and 0.27% are from ARG_2 to ARG_3. This type of cross-lingual role mismatch, also known as cross mapping, is also of particular interests since, if available, this knowledge source could be helpful

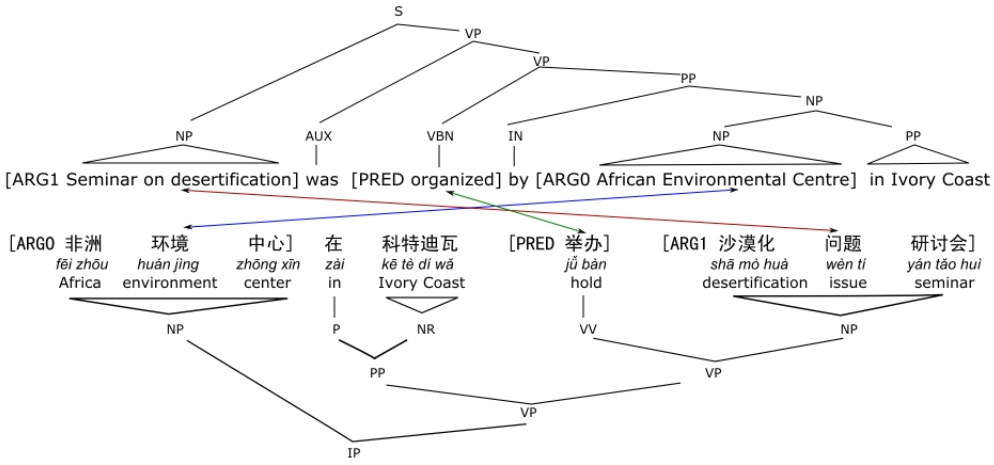


Figure 1: An example of bilingual semantic predicate argument mapping.

to MT systems.

One such cross-mapping example is shown below, where the "[$_{ARG1}$ world trade]" in English is mapped to "[$_{ARG0}$ 世界/world 贸易/trade]" in Chinese.

English Moreover , the report estimated that [$_{ARG1}$ world trade] [$_{ARGM-MOD}$ would] [$_{TARGET}$ grow] [$_{ARG2-EXT}$ by 9.4 %] [$_{ARGM-TMP}$ for 1997]

Chinese 此外， 报告 还 估计 [$_{ARGM-TMP}$ 1 9 9 7 年] [$_{ARG0}$ 世界 贸易] [$_{TARGET}$ 增长] [$_{ARG1}$ 百分之九点四]

Gloss Moreover, report also estimate 1997 year world trade grow 9.4%

4 ARG_ALIGN: Learning Bilingual Semantic Frames via Chinese/English Shallow Semantic Parsing

We propose to first use shallow semantic parsers to annotate Chinese and English bilingual sentences with their semantic role boundaries and labels. Next, we propose to align these predicate-argument structures in the bilingual sentences by an automatic mapping approach.

Given all the candidate semantic roles parsed from the automatic semantic parsers, the automatic role mapping problem is cast as follows:

$$Z^* := \sum_{i=1}^{n} \min_{x} \sum_{j=1}^{m} x_{ij} c_{ij} \quad (1)$$

$$s.t.$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, \cdots, m$$
$$x \geq 0$$

Z^* is the final role mappings we learned. x_{ij} is one element of the mapping matrix where argument i in Chinese is mapped to argument j in English, c_{ij} is one element of the cost matrix for aligning argument i in Chinese to j in English, n is the total number of arguments in a given source sentence and m is the total number of arguments in the target sentence.

To solve this bilingual predicate-argument role mapping problem, we propose an algorithm, ARG_ALIGN, as shown in Algorithm 1. In this algorithm, given S (source) and T (target) bi-sentence with semantic role annotation, we first match their predicate verbs based on a bilingual lexicon. Then, for each matched predicate verb pair *S-PRED* (source predicate) and *T-PRED* (target predicate), we extract their semantic arguments *S-ARG*s (source arguments) and *T-ARG*s (target arguments) and compute the cosine similarity score between all source and target arguments. We then extract the highest ranking matching pair of source and target constituents.

Algorithm 1 ARG_ALIGN

for each bilingual sentence pair **do**
  for each source predicate verb S-PRED **do**
    for each target predicate verb T-PRED **do**
      if S-PRED and T-PRED are translatable to each other, based on *bilingual lexicon* **then**
        S-ARGs $\Leftarrow ARG_0, \ldots, ARG_n$, given S-PRED
        T-ARGs $\Leftarrow ARG_0, \ldots, ARG_n$, given T-PRED
        for each ARG_i in S-ARGs **do**
          $\max(ARG_i) := 0$
          for each ARG_j in T-ARGs **do**
            align(ARG_i, $\hat{ARG}_j$)
            if $sim(ARG_i, ARG_j) \geq \max(ARG_i) \& sim(ARG_i, ARG_j) \geq threshold$ **then**
              $\max(ARG_i) := sim(ARG_i, ARG_j)$
              $\hat{ARG}_j := \text{argmax}\, ARG_i$
              where
              $sim(ARG_i, ARG_j) = \frac{ARG_i \cdot ARG_j}{|ARG_i||ARG_j|}$

4.1 Experimental Setup

Different sections of the Parallel Propbank corpus are used for algorithm development and evaluation. In order to determine the similarity threshold by which we can decide whether a pair of annotated bi-arguments match to each other, we randomly selected 497 sentence pairs as the test set and another set of 80 sentence pairs as the development data set.

Owing to the unavoidable errors through POS tagging, chunking or syntactic parsing, among the bilingual sentences, some Chinese and English sentences have no identifiable predicate verb, and are eliminated from further processing. Finally, 397 sentence pairs with automatic semantic parsing results are used in our predicate-argument mapping experiment.

In our proposed method, Chinese/English shallow semantic parsing is a prerequisite to achieving the task of bilingual semantic frame mapping. In recent years, there has been a lot of research on shallow semantic labeling or parsing both in English (Pradhan et al., 2004; Pradhan et al., 2005) and Chinese (Sun and Jurafsky, 2004; Xue and Palmer, 2005). In our experiments, we use the ASSERT semantic parser (Pradhan, 2005) to carry out the automatic semantic parsing on the English side and a similar SVM-based Chinese semantic parsing system (Wu et al., 2006) on the Chinese side. According to (Pradhan et al., 2005), their English semantic parser achieved 89.40 F-score with gold syntactic parse input, and 79.40 F-score with automatic syntactic parse input. Meanwhile, our SVM-based Chinese semantic parser yielded 89.89 F-score with gold syntactic parse input and 69.12 F-score with automatic syntactic parse input. Both of these parsers are among the-state-of-the-art shallow semantic systems in English and Chinese.

4.2 Experimental Results

Semantic role mapping output of our system is evaluated against the reference mappings described in the previous section, and measured with *Precision, Recall and F-score*[1] . In our evaluation strategy, a pair of arguments are considered correctly aligned to each other if the arguments are judged to be correct, and the mapping is judged to be correct.

The semantic role mapping result from our ARG_ALIGN algorithm is listed in Table 2 and the performance evaluation is listed in Table 3. 594 predicate-argument structure mappings are learned, with 219 unique Chinese verbs and 192 unique English verbs.

[1] F-score= $\frac{2 \times Precision \times Recall}{Precision + Recall}$

Table 2: Semantic Role Mappings from ARG_ALIGN

EN\CN	ARG0	ARG1	ARG2	ARG3
ARG0	259	8	7	0
ARG1	40	486	25	2
ARG2	3	26	15	0
ARG3	0	0	1	1

Table 3: Performance of Proposed Predicate-Argument Mapping

# words	[1,20]	<20,40>	[40,∞]	All
Precision	76.54	77.26	70.34	74.87
Recall	74.25	72.00	65.70	70.19
F-score	75.38	74.54	67.94	72.45

Many of these verbs are part of multiple context-dependent semantic structures. Human translation errors in the bilingual corpus, syntactic parsing and tagging errors account for some of the unmatched predicate-argument structures. Despite this, we obtained a fairly high F-score of 72.45% in bilingual semantic structure mapping, as evaluated against the mapping obtained from the oracle reference set.

5 Discussion of Results

Some of the mapping errors are due to errors in automatic syntactic and shallow semantic parsing. As a reference, we also evaluated the ARG_ALIGN algorithm directly on the Parallel Propbank data, by using the predicate-argument labels from manual annotation. The mapping accuracy in this case, free from parsing errors, is 98.9%.

Meanwhile, we observe that due to language differences and translation idiosyncrasies, predicate verb pairs in English versus Chinese do not always have the same argument structure. In this section, we present some interesting findings with examples in several categories.

5.1 Ellipsis

The ellipsis of some syntactic elements, such as the subject, occurred in either English or Chinese in the parallel sentences and might lead to some *NULL* argument mapping in the other language. As shown in the following example, [$_{ARG0}$ *PRO*] in Chinese is a filler constituent manually inserted in Chinese PropBank. However, the semantic role parser is not capable of generating this filler constituent automatically during the parsing. Thus, no *ARG*0 is labeled out in the automatic semantic parse result.

English Insiders feel that it would provide an excellent opportunity for [$_{ARG0}$ the economy and trade circles of China and South Korea] to [$_{TARGET}$ extend] [$_{ARG1}$ exchange and co-operation] .

Chinese 业内 人士 认为 ， 它 将 为 中 韩 两 国 经贸界 提供 一 次 [$_{ARG0}$ *PRO*] [$_{TARGET}$ 扩大] [$_{ARG1}$ 交流 与 合作] 的 良机 。

Gloss Inside people believe , it will be China Korea two country economy and trade circles provide a extend communication and co-operation excellent opportunity

5.2 Parallel Structures in Chinese

When a Chinese sentence consisting of a parallel structure is translated into English, the parallel structure is consistently translated to clauses in English since these syntactic alternations are an effective translation technique to represent the same meaning of Chinese in one English sentence. Argument mapping is nevertheless correct despite this type of syntactic mismatching, as shown in the following example.

English [$_{ARG1}$ An office of Shanghai Customs posted at Chongming] , that was [$_{TARGET}$ approved] [$_{ARG0}$ by the China Customs Head Office] [$_{ARG2}$ to be set up] , was established a few days ago , and has already officially conducted business.

Chinese 经 [$_{ARG0}$ 中国 海关 总署] [$_{TARGET}$ 批准] 设立 的 上海 海关 驻 崇明 办事处 于 日前 成立 ， 并 正式 对 外 办理 业务。

Gloss Via China Customs Headquarters approval establish Shanghai Customs station Chongming office in current set up, and officially conduct business .

5.3 One-to-many Role Mapping

In our proposed algorithm, role mapping is based on individual ARG, not the ARG combination. However, in reality, it is possible for there to be one-to-many mappings. Thus, when this occurs, the one-to-many mapping is not possible to be identified. For example, in the following bi-sentence, *ARG1* and *ARG2* in English are mapped to *ARG1* in Chinese together.

English At present , about 150 thousand foreign-invested enterprises have opened accounts in the Bank of China , of which , [$_{ARG0}$ more than 20 thousand enterprises] have [$_{TARGET}$ received] [$_{ARG1}$ loan support] [$_{ARG2}$ from the Bank of China] .

Chinese [$_{ARGM-TMP}$ 目前] ， 约 有 十五万 家 外商 投资 企业 在 中国 银行 开立 帐 户 ， 其中 [$_{ARG0}$ 二万多 家] [$_{TARGET}$ 获 得] [$_{ARG1}$ 中国 银行 的 贷款 支持] 。

Gloss currently, about 150 thousand foreign merchant investment enterprise in China Bank open account, of which, 20 thousand more enterprise receive China Bank's loan support .

6 Role Mapping from Syntactic Constituent Alignment

To date, it is often casually assumed that semantic roles can be simply projected across language pairs by constituent alignment (Pado and Lapata, 2006). In such an approach, it is assumed that an English constituent is lexically translated into the Chinese constituent, in which case they must share the same role label. This sort of view is typically inspired by the many structurally-based statistical machine translation models that make use of some kind of syntactic constituent projection (Hwa et al., 2005).

Therefore it is worth investigating the possibility of projecting semantic role labels across matching syntactic constituents. To accomplish this, we implement a contrastive SYN_ALIGN algorithm that obtains semantic structure mapping based on Treebank syntactic parse projection. This model is similar in spirit to that of (Pado and Lapata, 2006), in which the authors proposed a semantic role projection model based on FrameNet rather than PropBank verb frames. While our semantic role projection model is inspired by (Pado and Lapata, 2006), we propose a novel solution to the Linear Assignment Problem in order to align syntactic constituents from both the English and Chinese sentences, and then project the semantic role labels from English across to Chinese. The reason why we project the semantic role from English to Chinese is because according to (Pradhan et al., 2005), their English semantic parser outperforms our Chinese one due to the larger training data available in English TreeBank and PropBank.

In this approach, we make a strong assumption that the English semantic roles can be projected directly to their corresponding entities in Chinese (although, obviously, this assumption does not always hold in reality), and then utilize the lexical and syntactic information from the syntactic parses to project the semantic roles from English to Chinese.

To decouple the effect of semantic parsing from syntactic parsing, we save the syntactic annotations on the bilingual sentences, but remove the semantic annotations from the Chinese sentences. Based on the "perfect constituent alignment" proposed in (Pado and Lapata, 2006), we then project English semantic role labels to their corresponding Chinese entities. Finally, an evaluation of the mapping results are carried out in reference to the gold standard mapping set.

6.1 Alignment Selection

Since most structural machine translation systems are based on tree alignments, we are interested in investigating semantic role mapping on top of such syntax tree alignments. In other words, we select syntactic constituent (i.e. chunk) as the alignment unit. Moreover,

(Pado and Lapata, 2006) has also shown that the best semantic role projection is achieved with constituent based alignment.

6.2 Assignment Cost

Similar to (Pado and Lapata, 2006), we define the alignment cost between any pair of English and Chinese constituents as follows:

$$cost(e_c, c_c) = \frac{1}{sim(e_{c(w1,w2...)}, c_{c(w1,w2,...)})} \quad (2)$$

where, e_c is an English constituent, c_c is a Chinese constituent, w_i belongs to the set of $(NP, PP, pronoun, numeral, quantifier)$ and w_i is a content word. The purpose of this is to disregard any lexical items that would not be of interest to us in the ultimate task of argument mapping.

6.3 Constituent Alignment

(Pado and Lapata, 2006) proposed three alignment models for the constituent alignment: *total alignments*, *edge covers* and *perfect matchings*. We chose perfect matching for our experiment since (Pado and Lapata, 2006) reported superior performance using this model. "Perfect matching" is defined as follows: given all the constituents extracted from the Chinese and English parallel data, each constituent in Chinese must align to one and only one constituent in English, and vice versa. We observe that this problem can be cast as a Linear Assignment Problem, which of course is a fundamental combinatorial optimization problem. The Linear Assignment Problem can be described as follows:

$$Z^* := \min_x \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} c_{ij} \quad (3)$$

$$s.t.$$
$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, \cdots, n$$
$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, \cdots, n$$
$$x \geq 0$$

Z^* is the solution of the linear assignment problem. x_{ij} is the assignment matrix where constituent i was assigned to constituent j, c_{ij} is the cost matrix for aligning constituent i to j.

Table 4: Role Mapping from Syntactic Projection

EN\CN	ARG0	ARG1	ARG2	ARG3
ARG0	248	0	0	0
ARG1	0	381	0	0
ARG2	0	0	22	0
ARG3	0	0	0	0

Table 5: Performance of Semantic Role Projection

# words	[1,20]	<20,40>	[40,∞]	All
Precision	54.45	45.10	39.35	44.57
Recall	59.78	50.14	41.98	48.90
F-score	56.99	47.49	40.62	46.63

Our semantic role projection algorithm, SYN_ALIGN, is described in Algorithm 2. Given the English and Chinese bi-parse, we first extract their constituents (chunks). These constituents are stored in two arrays. Then, for these two constituent arrays, we apply the classic Hungarian method (Kuhn, 1955) to solve the Linear Assignment optimization problem by using the cosine similarity score between two constituents as the assignment cost. Finally, we project the English semantic roles to the Chinese side based on the constituent alignment result.

The predicate-argument mapping learned from the constituent based semantic role projection is shown in Table 4 and the performance evaluation against the mapping learned from the gold standard is shown in Table 5.

6.4 Experimental Results

Again evaluating with respect to the gold standard reference mappings, the mapping F-score of SYN_ALIGN is only 46.63%. This mapping performance is significantly lower than achieved by our proposed ARG_ALIGN model, owing to the assumption that argument structures can be projected across syntactic constituents, which has hereby been shown to be brittle.

Algorithm 2 SYN_ALIGN

INPUT: Chinese and English parallel syntactic parse trees
let $EN\_Cons[]$ = source English constituents
let $CN\_Cons[]$ = target Chinese constituents
en_no = number of English constituents
cn_no = number of Chinese constituents
$max\_no = maximum(cn\_no, en\_no)$
if $cn\_no < max\_no$ **then**
  append $max\_no - cn\_no$ with "dummy" constituents to $CN\_Cons[]$
else if $en\_no < max\_no$ **then**
  append $max\_no - en\_no$ with "dummy" constituents to $EN\_Cons[]$
for $i = 1$ to max_no **do**
  for $j = 1$ to max_no **do**
    $similarity\_score = cosine(CN\_Cons[i], EN\_Cons[j])$
    if $similarity\_score == 0$ **then**
      $cost\_matrix[i][j] = 1000.00$
    else
      $cost\_matrix[i][j] = 1/similarity\_score$
$alignment = hungarian\_method(cost\_matrix)$
for all semantic roles in English semantic parsing result **do**
  project the semantic roles to Chinese side based on *alignment* solution

7 Conclusion

For machine translation purposes, it is meaningful to study the semantic structural mapping between the source and target language. We propose a new automatic algorithm, ARG_ALIGN, to extract the predicate-argument mappings from unannotated bilingual sentence pairs with 72.45% F-score, given an unannotated parallel corpus. We first identify and label the semantic structures using the Chinese and English shallow semantic parsers and then use ARG_ALIGN to find the mapping pairs.

Given bilingual sentence pairs with manually annotated semantic role labels, we record the semantic role mapping between bilingual argument structures if they are *lexically* aligned to each other. We observe that there are 17.24% of cross mapping between argument structures in English and Chinese. Among these, 8.95% are argument 0-1 mappings, 6.94% are 1-2 mappings, and 0.27% are argument 2-3 mappings. Referring to the manual gold standard mapping, the F-score of our proposed mapping between automatically annotated argument structures is 72.45%, showing promise for automatic semantic structure mapping in bilingual sentence pairs, applicable to machine translation and other multilingual and cross-lingual applications.

Contrary to a preconception that one sometimes hears, we show empirically that our model is superior to a semantic role projection model which assumes semantic parallelism in bilingual sentences. In the latter model, we propose using the Hungarian method in a syntax alignment algorithm we name SYN_ALIGN, to align syntactic constituents from both the English and Chinese sentences, and project the semantic role labels across. Compared to the gold standard mapping, the mapping F-score in this case is 46.63%.

Our results led us to believe that, since there is a non-negligible amount of cross argument mapping between English and Chinese translations, it maybe preferable to use automatic semantic role labeling in both the source and target languages, than to use direct projection of semantic role labels from one language to the other.

One obvious next step is to embed the shallow semantic parsers and the cross-lingual verb frame acquisition model in end-to-end machine translation systems or MT applications. We would also like to acquire cross-lingual semantic frames for other categories besides verbs.

Acknowledgments

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under GALE Contract No. HR0011-06-C-0023, and by the Hong Kong Research Grants Council (RGC) research grant HKUST612806. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency.

We are grateful to Sebastian Pado, Martha Palmer, Sameer Pradhan, and Nianwen Xue for helpful discussions.

References

Marine Carpuat and Dekai Wu. 2005a. Evaluating the word sense disambiguation performance of statistical machine translation. In *Second International Joint Conference on Natural Language Processing (IJCNLP)*, pages 122–127, Jeju Island, Korea, Oct.

Marine Carpuat and Dekai Wu. 2005b. Word sense disambiguation vs. statistical machine translation. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, pages 387–394, Ann Arbor, Jun.

Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning (EMNLP-CoNLL 2007)*, Prague, Jun.

Marine Carpuat, Yihai Shen, Xiaofeng Yu, and Dekai Wu. 2006. Toward integrating word sense and entity disambiguation into statistical machine translation. In *Third International Workshop on Spoken Language Translation (IWSLT 2006)*, pages 37–44, Kyoto, Nov.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(03):311–325.

Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1):83–97.

Franz Josef Och, Daniel Gildea, Anoop Sarkar, Sanjeev Khudanpur, Kenji Yamada, Alexander Fraser, Libin Shen, Shankar Kumar, David Smith, Viren Jain, Katherine Eng, Zhen Jin, and Dragomir Radev. 2003. Website. `http://www.clsp.jhu.edu/ws2003/groups/translate/`.

Sebastian Pado and Mirella Lapata. 2006. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of ACL-COLING 2006*, pages 1161–1168, Sydney, Australia.

Martha Palmer, Nianwen Xue, Olga Babko-Malaya, Jinying Chen, and Benjamin Snyder. 2005. A parallel Proposition Bank II for Chinese and English. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 61–67, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of NAACL-HLT 2004*.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support vector learning for semantic argument classification. *Machine Learning*, 60(1):11–39.

Sameer Pradhan. 2005. ASSERT: Automatic Statistical SEmantic Role Tagger. Website. `http://oak.colorado.edu/assert/`.

Honglin Sun and Daniel Jurafsky. 2004. Shallow semantic parsing of Chinese. In *Proceedings of Human Language Technology Conference/North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

Dekai Wu and David Chiang, editors. 2007. *NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation (SSST)*. Association for Computational Linguistics, Rochester, NY, USA, April.

Zhaojun Wu, Yongsheng Yang, and Pascale Fung. 2006. C-ASSERT: Chinese shallow semantic parser. Website. `http://hlt030.cse.ust.hk/research/c-assert/`.

Nianwen Xue and Martha Palmer. 2005. Automatic semantic role labeling for Chinese verbs. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland*.

Capturing Translational Divergences with a Statistical Tree-to-Tree Aligner

Mary Hearne, John Tinsley, Ventsislav Zhechev and Andy Way

National Centre for Language Technology
Dublin City University
Dublin, Ireland
{mhearne,jtinsley,vzhechev,away}@computing.dcu.ie

Abstract

Parallel treebanks, which comprise paired source-target parse trees aligned at sub-sentential level, could be useful for many applications, particularly data-driven machine translation. In this paper, we focus on how translational divergences are captured within a parallel treebank using a fully automatic statistical tree-to-tree aligner. We observe that while the algorithm performs well at the phrase level, performance on lexical-level alignments is compromised by an inappropriate bias towards coverage rather than precision. This preference for high precision rather than broad coverage in terms of expressing translational divergences through tree-alignment stands in direct opposition to the situation for SMT word-alignment models. We suggest that this has implications not only for tree-alignment itself but also for the broader area of induction of syntax-aware models for SMT.

1 Introduction

Previous work has argued for the development of parallel treebanks, defined as bitexts for which the sentences are annotated with syntactic trees and are aligned below clause level (Volk and Samuelsson, 2004). Such resources could be useful for many applications, e.g. as training or evaluation corpora for word and phrase alignment, as training material for data-driven MT systems and for the automatic induction of transfer rules, and for translation studies. Their development is particularly pertinent to the recent efforts towards incorporating syntax into data-driven MT systems, e.g. (Melamed, 2004), (Chiang, 2005), (Galley et al., 2006), (Hearne and Way, 2006), (Marcu et al., 2006), (Zollmann and Venugopal, 2006).

In this paper, we focus on how translational divergences are captured within a parallel treebank using a fully-automatic statistical tree-to-tree aligner.[1] In doing so, we take a somewhat different perspective on tree-alignment from that of e.g. (Wu, 2000; Wellington et al., 2006). We do not incorporate trees for the express purpose of constraining the word- and phrase-alignment processes, although this is certainly a consequence of using trees. Our purpose in aligning monolingual syntactic representations is to make explicit the syntactic divergences between sentence pairs rather than homogenising them. We are not seeking to maximise the number of links between a given tree pair, but rather to find the set of links which most precisely expresses the translational equivalences between that tree pair. How best to exploit such information through model induction for syntax-aware statistical MT remains an open question.

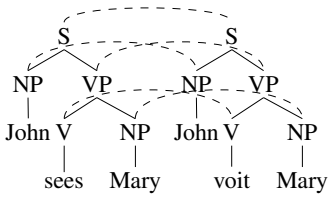
The remainder of this paper is organised as follows. In Section 2 we describe the tree-to-tree alignment process from a manual annotation per-

[1] Although the definition of a parallel treebank leaves room for a variety of types of tree structure, in this paper we focus on constituent structure trees only.

spective, outlining crucial ways in which it differs from the word-alignment process. We show how translational divergences are represented in an aligned parallel treebank in Section 3, giving insights into why such resources would be useful. In Section 4 we outline an automatic method for statistically inducing tree alignments between parsed sentence pairs – full details of the alignment algorithm are given in (Tinsley et al., 2007). In Section 5 we analyse the output to see how well translational divergences are captured. Finally, in Sections 6 and 7 we conclude and describe plans for future work.

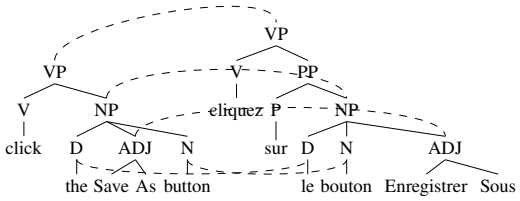
2 Manual Tree-to-Tree Alignment

The tree-to-tree alignment process assumes a parsed, translationally equivalent sentence pair and involves introducing links between non-terminal nodes in the source and target phrase-structure trees. Inserting a link between a node pair indicates that the substrings dominated by those nodes are translationally equivalent, i.e. that all meaning in the source substring is encapsulated in the target substring and vice versa. An example aligned English–French tree pair is given in (1). This example illustrates the simplest possible scenario: the sentence lengths are identical, the word order is identical and the tree structures are isomorphic.



(1)

However, most real-world examples do not align so neatly, as we will discuss in Section 3. The example given in (2) illustrates some important points. Not every node in each tree needs to be linked, e.g. *click* translates not as *cliquez*, but as *cliquez sur*. However, each node is linked at most once. Also, as we do not link terminal nodes, the lowest links are at the part-of-speech level. This means that multi-word units identified during parsing are preserved as such during alignment, cf. *Save As* and *Enregistrer Sous*.[2]



(2)

2.1 Tree Alignment vs. Word Alignment

When deciding how to go about linking a given tree pair, the logical starting point would seem to be with word alignment. However, some analysis reveals differences between the tasks of tree-alignment and word-alignment. We illustrate the differences by referring to the Blinker annotation guidelines (Melamed, 1998) which were used for the word alignment shared tasks at the workshops on *Building and Using Parallel Texts* at HLT-NAACL 2003[3] and ACL 2005.[4]

If a word is left unaligned in a sentence pair, it implies that the meaning it carries was not realised anywhere in the target string. On the other hand, if a node remains unaligned in a tree pair there is no equivalent implication. Because tree-alignment is hierarchical, many other nodes can carry indirect information regarding how an unaligned node (or group of unaligned nodes) is represented in the target string. Some consequences of this are as follows.

Firstly, the strategy in word-alignment is to leave as few words unlinked as possible "even when non-literal translations make it difficult to find corresponding words" (Melamed, 1998). Contrast this with the more conservative guideline for tree-alignment given in (Samuelsson and Volk, 2006): nodes are linked only when the substrings they dominate "represent the same meaning and ... could serve as translation units outside the current sentence context." This latter strategy is affordable because alignments at higher levels in the tree pair will account for the translation equivalence. Secondly, word-alignment allows many-to-many alignments at the word level but not phrasal alignments unless every word in the source phrase corresponds to every word in

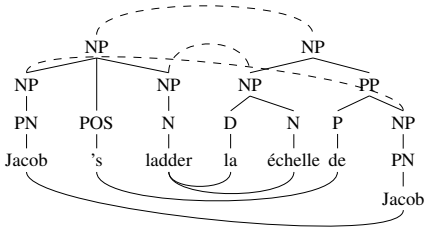
[2] Of course, an alternative parsing scheme which gives internal labelled structure in such phrases might permit further sub-tree links.

[3] http://www.cse.unt.edu/~rada/wpt/

[4] http://www.cse.unt.edu/~rada/wpt05/

the target and vice versa. Tree-alignment, on the other hand, allows each node to be linked only once but facilitates phrase alignment by allowing links higher up in the tree pair.

The contrasting effects of these guidelines are illustrated by the example given in (3)[5] where the dashed links represent tree-alignments and the solid links represent word-alignments. We see that the word-alignment must link *ladder* to both *la* and *échelle* whereas the tree-alignment specifies a single link between the nodes dominating the substrings *ladder* and *l'échelle*.

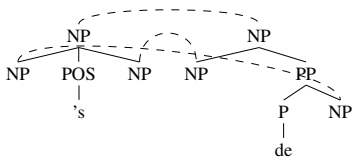


(3)

Note also that the word-alignment explicitly links *'s* with *de* whereas the tree-alignment does not; it is arguable as to whether these strings really represent precisely the same meaning. However, the relationship between these words is not ignored in the tree-alignment; rather it is captured by the link between the three NP links in combination.

In fact, many different pieces of information can be inferred from the tree-alignment given in (3) regarding the relationship between *'s* and *de*, despite the fact that they are not directly linked; examples exhibiting varying degrees of contextual granularity are given in (4).

$$
\begin{array}{ccc}
\textit{'s} & \longrightarrow & \textit{de} \\
\text{X } \textit{'s} \text{ Y} & \longrightarrow & \text{Y } \textit{de} \text{ X} \\
\text{NP}_1 \textit{ 's } \text{NP}_2 & \longrightarrow & \text{NP}_2 \textit{ de } \text{NP}_1 \\
\text{NP} \rightarrow \text{NP}_1 \textit{ 's } \text{NP}_2 & : & \text{NP} \rightarrow \text{NP}_2 \textit{ de } \text{NP}_1
\end{array}
\qquad (4)
$$



It is noteworthy, we feel, that the similarities between the 'rules' in (4) and templates in EBMT such as those in (Cicekli and Güvenir, 2003) are striking.

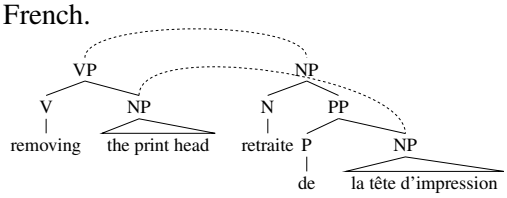
[5]The sentence pair and word alignments were taken directly from (Melamed, 1998).

3 Translational Divergences

Work such as that of e.g. (Lindop and Tsujii, 1992; Dorr, 1994; Trujillo, 1999) makes explicit the types of translational divergences which occur in real data. These divergences occur frequently even for language pairs with relatively similar surface word order, and generally prove challenging for MT models (Hutchins and Somers, 1992).[6] An important characteristic of parallel treebanks is that they provide explicit details, through tree-alignments, about the occurrence and nature of such divergences.

In this section, we examine how translational divergences are represented in the HomeCentre English–French parallel treebank. This corpus comprises a Xerox printer manual which was translated by professional translators and sentence-aligned and annotated at Xerox PARC. It contains 810 parsed, sentence-aligned English-French translation pairs. It was manually tree-aligned by one of the authors of this paper according to the guidelines outlined in Section 2.[7] As observed by (Frank, 1999), the HomeCentre corpus provides a rich source of both linguistic and translational complexity.

Instances of nominalisation are very frequent in the HomeCentre corpus. An example of a **simple nominalisation** is given in (5), where the English verb phrase *removing the print head* is realised as the noun phrase *retraite de la tête d'impression* in French.



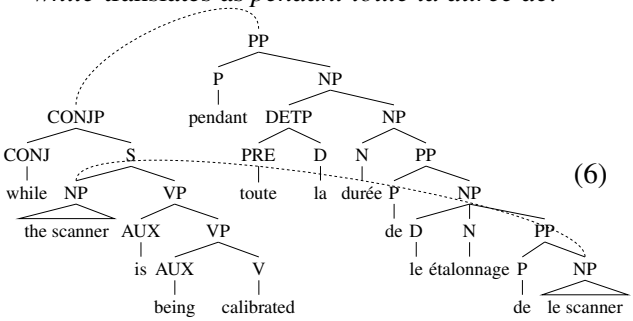
(5)

Instances of more **complex nominalisations** which incorporate further translational divergences are also common. Consider, for example, the translation pair given in (6). Firstly, we note the nominalisation: the English passive sentential form *the scanner is being calibrated* is realised as the French noun phrase *l'étalonnage*

[6]The picture is even more complex than we paint here; (Dorr et al., 2002) make the further observation that such 'hard' cases tend to co-occur much more often than might be expected.

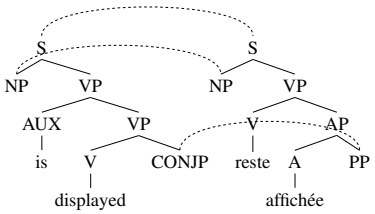
[7]As there was just a single annotator, inter-annotator agreement is obviously not a factor.

du scanner. However, we also observe the presence of **relation-changing**: the subject of this English sentential form, *the scanner*, functions as an oblique object in the French translation. In addition, this example exhibits stylistic divergence, as *while* translates as *pendant toute la durée de*.



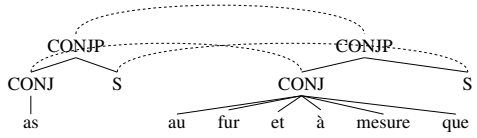
(6)

Another complex translation case which occurs in the HomeCentre corpus is that of **head-switching**, where the head word in the source language sentence translates as a non-head word in the target language realisation. An example of head-switching is given in (7). Here, the English verbal unit *is displayed* is realised in French as *reste affichée*; in this context, *reste* means (roughly) 'remains' and *display* is realised as the adverbial modifier *affichée*. Thus, the head of the English sentence, the verb *display*, corresponds to the French non-head word *affichée*.



(7)

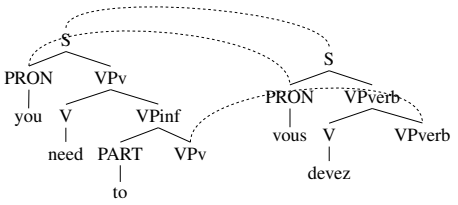
Of course, **lexical divergences** also occur frequently. In some instances, these divergences can be resolved in a straightforward manner. For example, we see in (8) that *as* in English can translate as *au fur et à mesure que* in French, but as the idiomatic reading of this French phrase is reflected in the parse assigned to the sentence, the overall shape of the sentence can remain the same despite the complexity of the translation.



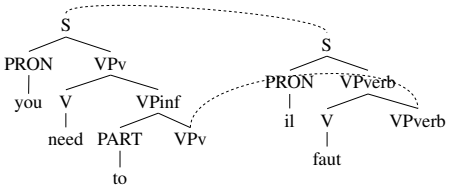
(8)

However, even for a relatively similar language pair, lexical divergence can cause source and target sentences expressing exactly the same concept to have completely **different surface realisations**. Consider, for example, the translation pair in Figure 1. As there is no French phrase which is directly equivalent to the English expression *null and void*, the given French sentence *toute intervention non autorisée invaliderait la garantie* – which translates roughly as 'any unauthorised action would invalidate the guarantee' – is entirely structurally dissimilar to its English counterpart.

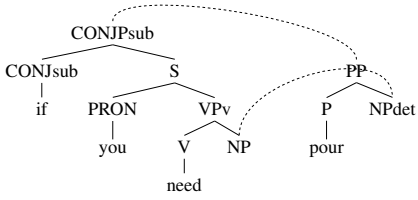
Finally, variation in how certain **frequently-occurring words** are translated, depending on the context in which the word appears, is also common. Examples (9) – (12) illustrate this phenomenon for the English verb *need*. *you need to X* can be realised as both *vous devez X* and *il faut X* in French, as shown in examples (9) and (10). The realisation differs, however, where the object is nominal rather than sentential: *if you need X* is shown in (11) to translate as *pour X*. Finally, we show in example (12) that the negative *you do not need to X* can translate as *il ne devrait pas être necessaire de X*, which literally means 'it should not be necessary to X' in English. We note that this is just a subset of the differing French realisations for the verb *to need* which occur in the HomeCentre corpus.



(9)



(10)



(11)

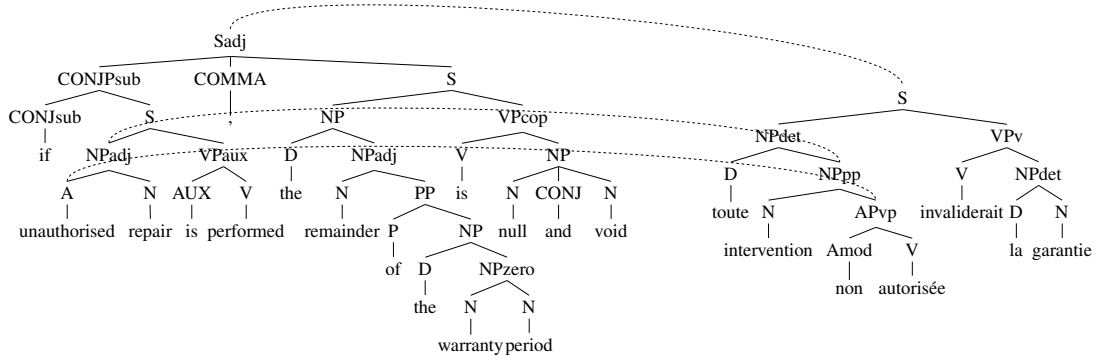
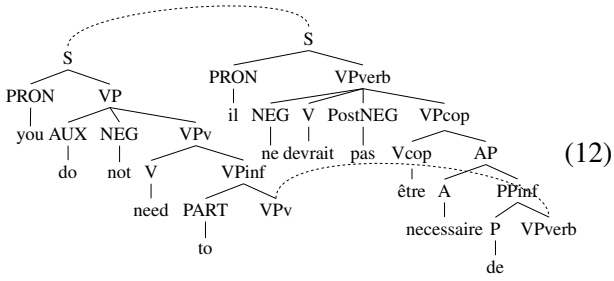


Figure 1: Completely different surface realisations can be seen even for language pairs with similar word order like English–French.



(12)

4 Automatic Tree-to-Tree Alignment

The tree-alignment algorithm briefly described here and detailed in (Tinsley et al., 2007) is designed to discover an optimal set of alignments between the tree pairs in a bilingual treebank while adhering to the following principles:

(i) independence with respect to language pair and constituent labelling schema;

(ii) preservation of the given tree structures;

(iii) minimal external resources required;

(iv) word-level alignments not fixed *a priori*.

4.1 Alignment Well-Formedness Criteria

Links are induced between tree pairs such that they meet the following well-formedness criteria:

(i) a node can only be linked once;

(ii) descendants of a source linked node may only link to descendants of its target linked counterpart;

(iii) ancestors of a source linked node may only link to ancestors of its target linked counterpart.

In what follows, a hypothesised alignment is ill-formed with respect to the existing alignments if it violates any of these criteria.

4.2 Algorithm

In this section we present how our alignment algorithm scores and selects links. We refer to the alternative methods by which decisions can be made at various points, and summarise the possible aligner configurations. (Tinsley et al., 2007) describes these variations in greater details and provides the motivation behind each variant.

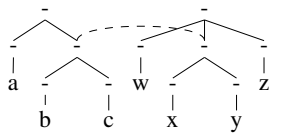
4.2.1 Selecting Links

For a given tree pair $\langle S, T \rangle$, the alignment process is initialised by proposing all links $\langle s, t \rangle$ between nodes in S and T as hypotheses and assigning scores $\gamma(\langle s, t \rangle)$ to them. All zero-scored hypotheses are blocked before the algorithm proceeds. The selection procedure then iteratively fixes on the highest-scoring link, blocking all hypotheses that contradict this link and the link itself, until no non-blocked hypotheses remain. These initialisation and selection procedures are given in **Algorithm 1** *basic*.

Algorithm 1 *basic*

Initialisation
  for each source non-terminal *s* **do**
    for each target non-terminal *t* **do**
      generate scored hypothesis $\gamma(\langle s, t \rangle)$
    end for
  end for
  block all zero-scored hypotheses
Selection *underspecified*
  while non-blocked hypotheses remain **do**
    link and block the highest-scoring hypothesis
    block all contradicting hypotheses
  end while

Hypotheses with equal scores: The **Selection** procedure given in **Algorithm 1** *basic* is incom-



s_l = b c
t_l = x y
$\overline{s_l}$ = a
$\overline{t_l}$ = w z

Figure 2: Values for s_l, t_l, $\overline{s_l}$ and $\overline{t_l}$ given a tree pair and a link hypothesis.

plete as it does not specify how to proceed if two or more hypotheses share the same highest score. When this case arises we invoke a method called *skip2*. Using this configuration, we skip over tied hypotheses until we find the highest-scoring hypothesis $\langle s, t \rangle$ with no competitors of the same score *and where neither s nor t has been skipped*.

Delaying lexical (span-1) alignments: It is sometimes the case that we want to delay the induction of lexical links in order to allow links higher up in the tree structures to be induced first. For this reason we have an optional configuration, *span1*. When this method is activated, it postpones links between any hypothesis $\langle x, y \rangle$, where either x or y is a constituent with a span of one, i.e. a lexical node. Only when all other possible hypotheses have been exhausted do we allow links of type $\langle x, y \rangle$.

4.2.2 Computing Hypothesis Scores

Inserting a link between two nodes in a tree pair indicates that (i) the substrings dominated by those nodes are translationally equivalent and (ii) all meaning carried by the remainder of the source sentence is encapsulated in the remainder of the target sentence. The scoring method we propose accounts for these indications.

Given tree pair $\langle S, T \rangle$ and hypothesis $\langle s, t \rangle$, we compute the following strings:

$$s_l = s_i ... s_{ix} \quad \overline{s_l} = S_1 ... s_{i-1} s_{ix+1} ... S_m$$
$$t_l = t_j ... t_{jy} \quad \overline{t_l} = T_1 ... t_{j-1} t_{jy+1} ... T_n$$

where $s_i ... s_{ix}$ and $t_j ... t_{jy}$ denote the terminal sequences dominated by s and t respectively, and $S_1 ... S_m$ and $T_1 ... T_n$ denote the terminal sequences dominated by S and T respectively. These string computations are illustrated in Figure 2.

The score for the given hypothesis $\langle s, t \rangle$ is computed according to (13).

$$\gamma(\langle s, t \rangle) = \alpha(s_l|t_l)\ \alpha(t_l|s_l)\ \alpha(\overline{s_l}|\overline{t_l})\ \alpha(\overline{t_l}|\overline{s_l}) \quad (13)$$

Individual string-correspondence scores $\alpha(x|y)$ are computed using word-alignment probabilities given by the Moses decoder[8,9] (Koehn et al., 2007). Two alternative scoring functions are given by *score1* (14) and *score2* (15).

Score *score1*

$$\alpha(x|y) = \prod_{j=1}^{|y|} \sum_{i=1}^{|x|} P(x_i|y_j) \quad (14)$$

Score *score2*

$$\alpha(x|y) = \prod_{i=1}^{|x|} \frac{\sum_{j=1}^{|y|} P(x_i|y_j)}{|y|} \quad (15)$$

4.3 Aligner Configurations

When configuring the aligner, we must choose *skip2* and we must choose either *score1* or *score2*. *span1* can be switched either on or off. The four possible configurations are as follows:

skip2_score1	skip2_score1_span1
skip2_score2	skip2_score2_span1

5 Alignment Evaluation and Analysis

In Section 5.1 we give an overview of aligner performance through two automatic evaluation methodologies. In Section 5.2 we then go on to describe the capture of translational divergences by manually analysing the aligner output.

5.1 Automatic Evaluation

We use two automatic evaluation methodologies in order to gain an overview of aligner performance: (i) we compare the links induced by the algorithm to those induced manually and compute precision and recall scores; (ii) we train a Data-Oriented Translation (DOT) system (Hearne and Way, 2006) on both the manually aligned data and the automatically aligned data and assess translation accuracy using the Bleu (Papineni et al., 2002), NIST (Doddington, 2002) and Meteor

[8] http://www.statmt.org/moses/

[9] Although our method of scoring is similar to IBM model 1, and Moses runs GIZA++ trained on IBM model 4, we found that using the Moses word-alignment probabilities yielded better results than those output directly by GIZA++.

	Alignment Evaluation						Translation Evaluation			
	all links		*lexical links*		*non-lexical links*		*(all links)*			
Configurations	Precision	Recall	Precision	Recall	Precision	Recall	Bleu	NIST	Meteor	Coverage
manual	–	–	–	–	–	–	0.5222	6.8931	71.8531	68.5417
skip2_score1	0.6162	0.7783	0.5057	0.7441	**0.8394**	0.7486	0.5091	6.9145	71.7764	71.8750
skip2_score2	0.6215	0.7876	0.5131	0.7431	0.8107	0.7756	**0.5333**	6.8855	**72.9614**	**72.5000**
skip2_score1_span1	**0.6256**	**0.8100**	0.5163	**0.7626**	0.8139	**0.8002**	0.5273	**6.9384**	72.7157	**72.5000**
skip2_score2_span1	0.6245	0.7962	**0.5184**	0.7517	0.8031	0.7871	0.5290	6.8762	72.8765	**72.5000**

Table 1: Evaluation of aligner performance using automatic metrics.

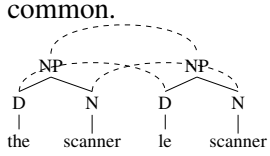
(Banerjee and Lavie, 2005) automatic evaluation metrics. The results of these evaluations are given in Table 1.

Looking firstly at overall alignment accuracy (the *all links* column), it is immediately apparent that recall is significantly higher than precision for all configurations. In fact, we have observed that all aligner variations consistently induce more links than exist in the manual version, with the average number of links per tree pair ranging between 10.4 and 11.0 for the automatic alignments versus 8.3 links per tree pair for the manual version. A clearer picture emerges when we differentiate between lexical and non-lexical links, where a link is non-lexical if both source and target nodes span more than one terminal. We see that, actually, precision is higher than recall for non-lexical links, and overall accuracy is higher for non-lexical links than for all links. In contrast, overall accuracy is much lower for lexical links than for all links, and the disparity between precision and recall is greater.

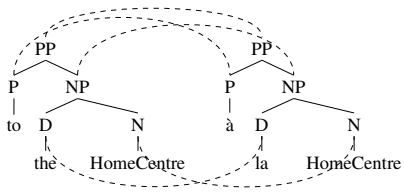
Turning our attention to translation accuracy, we observe that the scores for the automatic alignments are very encouraging: for all three evaluation metrics, at least two aligner configurations outperform the manual scores. Furthermore, all the automatically-aligned datasets achieve higher coverage than the manually-aligned run. It is perhaps somewhat surprising that the translation scores do not reflect the indication given by the alignment evaluation that word-level alignment precision is low compared to phrase-level precision. The explanation as to why the translation scores do not deteriorate may lie in how the MT system works: because DOT displays a preference for using larger fragments when building translations wherever possible, the impact of inconsistencies amongst smaller fragments (i.e. word-level alignments) is minimised. The reason for the improvement in scores lies in the increased coverage of the system trained on the automatic alignments.

5.2 Capturing translational divergences

Before looking at divergent cases, we first observe that the alignment algorithm generally produces accurate output for the simple translation cases. Examples (16) and (17) illustrate cases where the aligner correctly identifies equivalent constituents where length, word order and tree structure all match perfectly. For short phrases, such examples are relatively common.



(16)

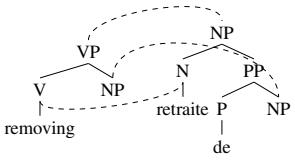


(17)

Lexical divergences which are of the form 1-to-many and many-to-1 occur frequently in the data and the aligner captures them with regularity. For example, the aligner output exactly matches the manual alignment for example (8). As mentioned in Section 4, when calculating the score for a particular hypothesis, we not only consider the translational equivalence of the dominated substrings but also the translational equivalence of the remainder of the source and target sentences. In this way, links can be inferred even when the constituent substrings are lexically divergent.

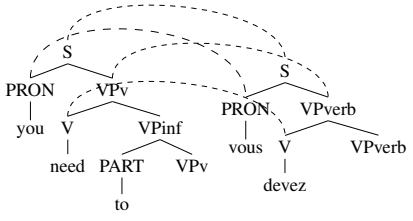
Instances of nominalisation are also commonly presented to the aligner. Consider, for example, the aligner output in (18) where the English verb phrase *removing the print head* is re-

alised as the French noun phrase *retraite de la tête d'impression*. As the aligner does not take into consideration the labels on the tree, but rather the likelihood that the surface strings are translations of each other, there is no impediment to the linking of the English VP to the French NP. Furthermore, the lower NP alignment is straightforward. Note, however, the (probably incorrect) link between the V *removing* and the N *retraite*. This link does not appear in the manual alignment (shown in (5)) as the annotator considered the meaning equivalence to be between *removing* and *retraite de*.

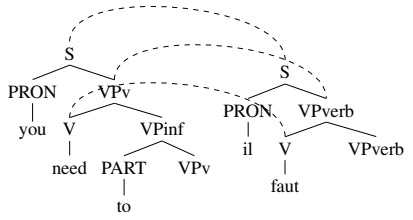


(18)

In Section 3 we noted that frequently-occurring words vary greatly in terms of how they are translated, as illustrated for the English verb *need* in examples (9) – (12). These examples are handled reasonably well by the aligner, again due to the strength of the equivalence between the object constituents. In (19) and (20) (for which the manual alignments were given in (9) and (10)), we again see lexical alignments in the automatic output which were not included in the manual versions; the annotator considered the equivalences to be (*need to*, *devez*) and (*you need to*, *il faut*). While the case for linking *need* with *devez* is arguable, the link between *need* and *faut* is incorrect.



(19)



(20)

The relation-changing and head-switching cases illustrated by (6) and (7) are not handled correctly by the aligner. However, in both cases poor choice of lexical alignments (for *being* and *reste* respectively) ruled out the possibly of correct higher-level alignments. Whether improved lexical choice will lead to the identification of the appropriate alignments in these cases remains to be seen.

6 Conclusions

We observe that while the algorithm performs well at the phrase level, performance on lexical-level alignments is relatively poor when we compare the aligner output to the manual alignments. This can be seen both in terms of precision and recall, where scores for phrase-level alignments are much higher than those for lexical ones, and through the manual evaluation where complex translation phenomena are identified correctly at a high level but then negated by inaccurate alignments at lexical level.

The lexical accuracy scores illustrate clearly that there is an imbalance between precision and recall: recall is consistently higher than precision across all variants of the alignment algorithm. The reason for this is based in the word-alignments used to seed our tree-alignment algorithm. We have adopted the widely used alignment tool GIZA++ (Och and Ney, 2003) (and, more recently, Moses (Koehn et al., 2007) which is based directly on GIZA++) which prioritises broad coverage rather than high precision (Tiedemann, 2004) and is appropriate to string-based SMT (Koehn et al., 2003). However, the work presented here indicates that the preference in terms of expressing translational divergences through tree-alignment is for the opposite – high precision rather than broad coverage – and this mismatch appears to impact on the overall quality of the alignments. We suggest that this has implications not only for tree-alignment itself but also for the broader area of induction of syntax-aware models for SMT.

Despite these observations, training our DOT system on automatically-aligned data gives slightly better translation performance than training on the manually-aligned data. The issue of coverage is key here. Crucially, the only model used by the system is the synchronous tree-substitution grammar induced directly from the parallel treebank. As the manual alignments con-

tain fewer links than the automatic alignments, the induced grammar achieves correspondingly lower coverage and, consequently, performance suffers. We conclude that it is appropriate for tree-alignment to prioritise precision in order to capture translational divergences as accurately as possible, and that MT systems making use of these alignments should employ them in conjunction with broad-coverage models (such as word- and phrase-alignments) in order to preserve robustness.

7 Future Work

In order to improve the accuracy of our tree-alignment algorithm, we plan to investigate alternative word-alignment techniques (e.g. (Tiedemann, 2004; Liang et al., 2006; Ma et al., 2007)) in order to establish which one is most appropriate for our task.

With regard to the broader area of parallel treebank construction and the use of statistical parsers such as those of Charniak (2000) and Bikel (2002), we would like to examine the impact of imperfect parse quality on the capture of translational divergences. We plan to extend our aligner so that it works with n-best parse forests on the source and/or target sides, thereby giving the aligner some (limited) influence over the configuration of the aligned parse trees.

Finally, we plan to investigate how best to incorporate the translation information encoded in parallel treebanks into existing data-driven MT systems, both indirectly in terms of complementary phrase/chunk extraction methods and directly in terms of inducing syntactic models of translation.

Acknowledgements

This work was generously supported by Science Foundation Ireland Grant No. 05/RF/CMS064 and the Irish Centre for High-End Computing.[10] We thank Khalil Sima'an, Declan Groves and the anonymous reviewers for their insightful comments.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43th Annual Meeting of the Association of Computational Linguistics (ACL-05)*, pages 65–72, Ann Arbor, MI.

Daniel M. Bikel. 2002. Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine. In *Proceedings of the 2nd International Conference on Human Language Technology Research*, pages 24–27, San Francisco, CA.

Eugene Charniak. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of the 1st Conference on North American Chapter of the Association for Computational Linguistics*, pages 132–139, Seattle, Washington.

David Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, MI.

Ilyas Cicekli and H. Altay Güvenir. 2003. Learning Translation Templates from Bilingual Translation Examples. In Michael Carl and Andy Way, editors, *Recent Advances in Example-Based Machine Translation*, pages 255–286. Kluwer Academic Publishers, Dordrecht, The Netherlands.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Human Language Technology: Notebook Proceedings*, pages 128–132, San Diego, CA.

Bonnie J. Dorr, Lisa Pearl, Rebecca Hwa, and Nizar Habash. 2002. DUSTer: A Method for Unravelling Cross-Language Divergences for Statistical Word-Level Alignment. In *Machine Translation: From Research to Real Users. Proceedings of the 5th Conference of the Association for Machine Translation in the Americas (AMTA 2002)*, pages 31–43, Tiburon, CA.

Bonnie J. Dorr. 1994. Machine translation divergences: a formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.

Anette Frank. 1999. LFG-based syntactic transfer from English to French with the Xerox Translation Environment. In *Proceedings of the ESSLLI'99 Summer School*, Utrecht, The Netherlands.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable Inference and Training of Context-Rich Syntactic Translation Models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia.

[10] http://www.ichec.ie/

Mary Hearne and Andy Way. 2006. Disambiguation Strategies for Data-Oriented Translation. In *Proceedings of the 11th Conference of the European Association for Machine Translation (EAMT-06)*, pages 59–68, Oslo, Norway.

W. John Hutchins and Harold Somers. 1992. *An Introduction to Machine Translation*. Academic Press, London.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03)*, pages 48–54, Edmonton, Canada.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, pages 177–180, Prague, Czech Republic.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '06)*, pages 104–111, New York City, NY.

Jeremy Lindop and Jun-ichi Tsujii. 1992. Complex Transfer in MT: A Survey of Examples. Technical Report 91-5, Centre for Computational Linguistics, UMIST, Manchester.

Yanjun Ma, Nicolas Stroppa, and Andy Way. 2007. Bootstrapping Word-Alignment via Word Packing. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, pages 304–311, Prague, Czech Republic.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical Machine Translation with Syntactified Target Language Phraases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP-06)*, pages 44–52, Sydney, Australia.

I. Dan Melamed. 1998. Annotation Style Guide for the Blinker Project. Technical Report 98-06, IRCS, University of Pennsylvania, Philadelphia, PA.

I. Dan Melamed. 2004. Statistical Machine Translation by Parsing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, pages 653–660, Barcelona, Spain.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 311–318, Philadelphia, PA.

Yvonne Samuelsson and Martin Volk. 2006. Phrase Alignment in Parallel Treebanks. In *Proceedings of the 7th Conference of the 5th Workshop on Treebanks and Linguistic Theories (TLT 2006)*, Prague, Czech Republic.

Jörg Tiedemann. 2004. Word to Word Alignment Strategies. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, pages 212–218, Geneva, Switzerland.

John Tinsley, Ventsislav Zhechev, Mary Hearne, and Andy Way. 2007. Robust Language Pair-Independent Sub-Tree Alignment. In *MT Summit XI*, Copenhagen, Denmark.

Arturo Trujillo. 1999. *Translation Engines*. Springer, London.

Martin Volk and Yvonne Samuelsson. 2004. Bootstrapping Parallel Treebanks. In *Proceedings of the 7th Conference of the Workshop on Linguistically Interpreted Corpora (LINC)* , pages 71–77, Geneva, Switzerland.

Benjamin Wellington, Sonjia Waxmonsky, and I. Dan Melamed. 2006. Empirical Lower Bounds on the Complexity of Translational Equivalence. In *Proceedings of the 44th annual conference of the Association for Computational Linguistics (ACL-06)*, pages 977–984, Sydney, Australia.

Dekai Wu. 2000. Bracketing and aligning words and constituents in parallel text using Stochastic Inversion Transduction Grammars. In Jean Veronis, editor, *Parallel Text Processing: Alignment and Use of Translation Corpora*, pages 139–167. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings of the HLT-NAACL 2006 Workshop on Statistical Machine Translation*, pages 138–141, New York City, NY.

Extracting Phrasal Alignments from Comparable Corpora by Using Joint Probability SMT Model

Tadashi Kumano[†,‡] **Hideki Tanaka**[†]
[†]NHK Science and Technical Research Laboratories
Tokyo, JAPAN 157-8510
{kumano.t-eq,tanaka-h.ja}@nhk.or.jp

Takenobu Tokunaga[‡]
[‡]Department of Computer Science
Tokyo Institute of Technology
Tokyo, JAPAN 152-8552
take@cl.cs.titech.ac.jp

Abstract

We propose a method of extracting phrasal alignments from comparable corpora by using an extended phrase-based joint probability model for statistical machine translation (SMT). Our method does not require preexisting dictionaries or splitting documents into sentences in advance. By checking each alignment for its reliability by using log-likelihood ratio statistics while searching for optimal alignments, our method aims to produce phrasal alignments for only parallel parts of the comparable corpora. Experimental result shows that our method achieves about 0.8 in precision of phrasal alignment extraction when using 2,000 Japanese-English document pairs as training data.

1 Introduction

Comparable corpora as a source of translation knowledge have attracted the attention of many researchers. Comparable corpora are composed of document pairs describing the same topic in different languages. They are not *parallel* (mostly word-to-word translated) corpora composed of good bilingual sentence pairs, but still contain various levels of parallelism, such as words, phrases, clauses, sentences, and discourses, depending on the corpora characteristics. Compared with parallel corpora, comparable corpora are much easier to build from commonly available documents, such as news article pairs describing the same event in different languages.

Recently, many studies on automatic acquisition of parallel parts from noisy non-parallel corpora have been conducted to acquire larger training corpora for statistical machine translation (SMT). One of the recent studies tried to find parallel sentences (Zhao and Vogel, 2002; Munteanu and Marcu, 2002; Fung and Cheung, 2004), and another tried to extract sub-sentential parallel fragments (Munteanu and Marcu, 2006). To detect the parallel parts, most of these studies required good statistical bilingual dictionaries, which are extracted from parallel corpora. Here we face "the chicken or the egg" problem. Previous studies use preexisting parallel corpora as bootstraps to prepare dictionaries, but it would be better to obtain lexical translation knowledge and extract parallel parts (eliminate unrelated parts) from comparable corpora simultaneously without parallel corpora.

In this paper, we propose an extension of the phrase-based joint probability model for SMT proposed by Marcu and Wong (2002). Our method can extract phrase alignments directly from comparable document pairs, without preexisting dictionaries or preprocessing of training data such as splitting it into sentences or extracting parallel parts. To prevent from producing alignments between unrelated phrases while searching for optimal alignments, we check each alignment as to

Original Japanese script:	Script translated into English:
1: 地震が続いている伊豆諸島できょう午前六時四十二分頃強い地震があり式根島で震度五弱を観測しました。 (There was a strong earthquake in the Izu Islands at 6:42 this morning, and the quake was measured the intensity of five-minus on the Japanese scale of seven at Shikine Island. A series of earthquakes have recently occurred around Izu Islands.)	1: A strong earthquake jolted Shikine Island, one of the Izu islands south of Tokyo, early on Thursday morning.
2: このほか震度四が新島、神津島、震度三が利島、三宅島、また関東各地や静岡県の一部で震度二や一の揺れを観測しました。 (The measurements of the quake at other places are as follows: intensities of four at Niijima and Kozu Islands, three at Toshima and Miyake Islands, and two or one at several places in the Kanto area and a part of Shizuoka Prefecture.)	2: The Meteorological Agency says the quake measured five-minus on the Japanese scale of seven.
3: この地震による津波の心配はありません。 (Official says there will be no fear of tsunamis caused by this earthquake.)	3: The quake affected other islands nearby.
4: 気象庁の観測によりますと震源地は新島・神津島の近海で震源の深さは十キロ、地震の規模を示すマグニチュードは五点一と推定されています。 (According to the observation of the Meteorological Agency, the center of the earthquake was 10 kilometers under the the sea bottom near Niijima and Kozu Islands, and the magnitude was 5.1.)	4: Seismic activity began in the area in late July, and 17 quakes of similar or stronger intensity have occurred.
5: 六月末から地震活動が始まった伊豆諸島では活動が活発な状態とやや落ち着いた状態を繰り返していて、先月三十日も三宅島で震度六弱の強い地震を一回観測した他震度五強の地震が二回起きました。 (Intermittent seismic activity began in the Izu Islands in late July, and the recent quakes were observed on the 30th of last month, once with an intensity of six-minus at Miyake Island and twice with an intensity of five-minus nearby.)	5: Officials are warning of more similar or stronger earthquakes around Niijima and Kozu Islands.
6: これらの地震を含めて一連の地震活動では神津島や新島、三宅島で震度六弱の強い揺れを四回観測したのを含めてこれまでに震度五弱以上の地震が十七回起きています。 (17 quakes with intensities of five-minus or higher including the recent ones have occurred during the activity, including four strong quakes with intensities of six-minus observed at Kozu, Niijima and Miyake Islands.)	6: Tokyo police say there have been no reports of damage from the latest quake.

Figure 1: Example article pair from the NHK Japanese-English news corpus

whether it is a statistically reliable translation by using log-likelihood ratio (LLR) statistics. The experimental results on our extension of Marcu-Wong's Model 1 shows that it is effective for extracting phrase alignments from comparable corpora. Those phrasal alignments are useful in applications other than machine translation. For example, we are developing a comparable translation retrieval system for supporting professional translators. The system will be more effective if it is able to show how a part in a source document is translated in a counterpart in response to the user's requests.

Section 2 introduces the Japanese-English broadcast news corpus, which is the target of our proposing method, and explains our tasks. Section 3 explains our improvements to the phrase-based joint probability model of Marcu and Wong in order to apply it to comparable corpora. After that, we show the results of our preliminary alignment experiment and discuss the effectiveness of our method in Section 4. Section 5 refers to related works and Section 6 concludes our paper.

2 Alignment Task for NHK Japanese-English News Corpus

We have been studying possible alignment methods for our comparable corpus, the NHK

Japanese-English news corpus, which is composed of pairs of Japanese news scripts and their manual translations into English broadcasted by NHK (Japan Broadcasting Corporation)[1]. The articles in Japanese and English in our corpus respectively have about 5 and 8 sentences on average.

An example article pair is shown in Figure 1 (The Japanese article is provided with a literal English translation for convenience). This example shows that the article pair shares the same topic, but each article describes the topic in a different style. Some articles have partially different content from their counterparts. Therefore, few parallel sentence pairs can be found in this corpus. At the level of words or shorter collocations, many useful translations can be found. However, words or phrases in a sentence are often translated into different sentences in the counterpart language. Thus, if you estimate word or phrase alignments from this type of comparable corpora, you have to search the whole document of the counterpart language.

3 Extension of Phrase-Based Joint Probability Model

Marcu and Wong (2002) proposed a joint probability model. It models how source and target sentences are simultaneously generated by *concepts*. Many of the phrase-based SMT models require word-level alignments for extracting phrases from combinations of the alignments. On the other hand, their training method can learn word and phrase alignments at the same time for searching for optimal alignments among possible partial word sequences in sentence pairs. There was a report that the joint probability model achieved better performance on SMT, especially for small-sized training data (Birch et al., 2006).

The formulation of Marcu-Wong model can be simply extended to non-parallel corpora by adding a means of handling monolingual phrases appearing independently of any counterpart. The search for optimal phrase alignments in their training method can be straightforwardly viewed as finding the parallel parts in a comparable document pairs. Therefore, we choose to employ their joint probability model for comparable corpora.

The main difficulty of the extension is the arbitrariness of deciding how many portions in each of the document pairs should be considered as unrelated to the counterpart document. We try to resolve the difficulty with the help of the log-likelihood ratio statistics to distinguish reliably correlated translations from unrelated parts.

3.1 Model Formulation

The original joint probability model assumes that every part of the sentences on the source and target sides is composed of phrases generated from *concepts*. We extended the model so that comparable document pairs have not only parallel phrases that share *concepts* but also non-parallel phrases that are independent of the counterpart document.

We consider a concept so that they can generate a monolingual phrase only on either side of a document pair. Under this definition, we can use the following formula, which is the same as the Marcu-Wong method, to express the probability of generating a document pair $(\boldsymbol{e}, \boldsymbol{f})$ which may have non-parallel phrases:

$$p(\boldsymbol{e}, \boldsymbol{f}) = \sum_{C \in \{\mathcal{C} | L(\boldsymbol{e}, \boldsymbol{f}, C)\}} \prod_{c_i \in C} t(\vec{e}_i, \vec{f}_i), \quad (1)$$

where

$\vec{e}$, $\vec{f}$: source and target phrases which are empty ( ) or consist of sequences of one or more words,

c_i: a concept to generate a pair of source and target phrases $(\vec{e}, \vec{f})$ only one side of which can be . Each concept produces a unique pair of phrases (or a monolingual phrase), so we indicate a concept as a pair of phrases like $(\vec{e}, \vec{f})$.

In this model, a document pair can be linearized with various degrees of parallelness from completely independent (when every c_i is monolingual) to completely parallel (when every c_i is bilingual).

[1] http://www.nhk.or.jp/english/

3.2 Training Procedure

Our training procedure consists of the following steps similar to those of the Marcu-Wong method:

1. Initialize distributions.
2. For each document pair, produce an initial alignment by linking phrases so as to create bilingual or monolingual concepts that have high t for all words in the document pairs. Then hillclimb towards the Viterbi alignment by breaking and merging concepts, swapping words between concepts, and moving words across concepts, so as to maximize the product of t.
3. Update distributions with the results of hillclimbing in step 2.
4. Iterate step 2.–3. several times.

We use a suffix array data structure for counting phrase occurrences (Callison-Burch et al., 2005), so we don't need to select only the limited number of high-frequency n-grams as phrase candidates.

In the following sections we give a detailed explanation of our extensions to the steps of the Marcu-Wong method.

3.2.1 Initializing Distributions

***t*-distribution** We define a phrase as a continuous sequence of zero or more words which does not extend more than one sentence. Under this definition, a document consisting of w words and s non-empty sentences can be partitioned into i non-empty phrases in $\binom{w-s}{i-s}$ ways, because the document has $w \quad s$ partitionable word boundaries and $i \quad s$ times of partitioning makes s pieces into i fragments[2]. Given that any phrases in $\boldsymbol{e}$ consisting of w_e words and s_e non-empty sentences can be mapped to any phrase in $\boldsymbol{f}$ consisting of w_f words and s_f non-empty sentences, there are $A(w_e, s_e, w_f, s_f)$ ways of alignments that can be built between $(\boldsymbol{e}, \boldsymbol{f})$:

$$A(w_e, s_e, w_f, s_f) = \sum_{k=0}^{\min(w_e, w_f)} k! \sum_{i=\max(k, s_e)}^{w_e} \sum_{j=\max(k, s_f)}^{w_f} \binom{w_e \quad s_e}{i \quad s_e} \binom{i}{k} \binom{w_f \quad s_f}{j \quad s_f} \binom{j}{k}. \quad (2)$$

In this formula, k denotes the number of bilingual concepts that $(\boldsymbol{e}, \boldsymbol{f})$ shares, and i and j denote the number of phrases which $\boldsymbol{e}$ and $\boldsymbol{f}$ are partitioned into, which follows that $\boldsymbol{e}$ and $\boldsymbol{f}$ have $i \quad k$ and $j \quad k$ phrases generated from monolingual concepts, respectively.

When the EM training starts without any information, all of the $A(w_e, s_e, w_f, s_f)$ alignments that can be built between the document pair $(\boldsymbol{e}, \boldsymbol{f})$ can be assumed to occur with the same probability. Under these conditions, the probability that a bilingual concept $(\vec{e}, \vec{f})$ occurs to generate non-empty phrases $\vec{e}$ and $\vec{f}$ consisting of l_e and l_f words in the document pair $(\boldsymbol{e}, \boldsymbol{f})$ is

$$\frac{A(w_e \quad l_e, s_e + \delta_e, w_f \quad l_f, s_f + \delta_f)}{A(w_e, s_e, w_f, s_f)}. \quad (3)$$

If $\vec{e}$ is placed in the middle of a sentence so that its removal separates the sentence into two non-empty parts, then $\delta_e = 1$; if $\vec{e}$ shares a single end with a sentence so that its removal from the sentence leaves a single non-empty sequence, then $\delta_e = 0$; and if $\vec{e}$ covers the whole of a sentence, then $\delta_e = \quad 1$ (δ_f likewise).

Similarly, the probability that a monolingual concept $(\vec{e}, \quad)$ occurs to generate a non-empty phrase $\vec{e}$ consisting of l_e words in the document pair $(\boldsymbol{e}, \boldsymbol{f})$ is:

$$\frac{A(w_e \quad l_e, s_e + \delta_e, w_f, s_f)}{A(w_e, s_e, w_f, s_f)} \quad (4)$$

(and likewise for concept $(\quad, \vec{f})$).

We can consider the probabilities (3) and (4) for each concept as the expected counts for which the concept contributes to the generation of the document pairs. We collect these counts for each document pair in a corpus,

[2] Although it is not theoretically essential to do so, we strictly enumerate the ways of partitioning, unlike in the Marcu-Wong method which approximates them by using the Stirling number.

and then obtain an initial joint distribution t by normalizing the counts to obtain probabilities. The use of a suffix array data structure for counting phrases enables us to calculate each t probability on the fly while EM training without a prepared table. The only thing we have to calculate beforehand is the total counts as a normalization factor.

***o*-distribution** In addition to the t-distribution, we need a distribution of phrase cooccurrence counts o, for checking the correlation between the bilingual phrase pairs described in the next section.

We consider a pair of bilingual phrases $\vec{e}$ and $\vec{f}$ in a document pair $(\boldsymbol{e}, \boldsymbol{f})$ to be cooccurring phrases if they are potentially generable by a bilingual concept; i.e. the pair is generated by a bilingual concept, or each of the pair is separately generated by a monolingual concept. In addition, we assume that only smaller number of cooccurrences between a and b are observed when $\vec{e}$ (we call each of them $\vec{e_1}, \ldots, \vec{e_a}$) in $\boldsymbol{e}$ appears a times and $\vec{f}$ (we call each of them $\vec{f_1}, \ldots, \vec{f_b}$) in $\boldsymbol{f}$ appears b times. There are $(a + \sum_{n=1}^{b-1} \sum_{c=1}^{a-n} c)$ ways of alignments between $(\boldsymbol{e}, \boldsymbol{f})$ where the same number of $\vec{e}$ and $\vec{f}$ are generated from monolingual concepts in each side of the document pair (assuming $a > b$), so the cooccurrence counts for a pair $(\vec{e}, \vec{f})$ cooccurring in $(\boldsymbol{e}, \boldsymbol{f})$ can be calculated as follows:

$$\left(1 + \frac{a + \sum_{n=1}^{b-1} \sum_{c=1}^{a-n} c}{ab}\right) \sum_{i=1}^{a} \sum_{j=1}^{b} \frac{A(w_e \quad l_e, s_e + \delta_{e_i}, w_f \quad l_f, s_f + \delta_{f_j})}{A(w_e, s_e, w_f, s_f)}. \tag{5}$$

We collect the counts of each document pair in a corpus to obtain the initial cooccurrence distribution o. As in the calculation of the t-distribution, we only need to prepare the total counts before EM training.

3.2.2 Producing Alignments with Log-Likelihood Ratio (LLR) Checking

To produce the alignments in step 2, we statistically check the bilingual concepts by using log-likelihood ratio (LLR) statistics (Dunning, 1993) so as to produce only concepts of reliably correlated phrase pairs (Moore, 2004; Munteanu and Marcu, 2006). Note that monolingual concepts are all available without checking. The checking procedure for a concept $(\vec{e}, \vec{f})$ is as follows:

1. Prepare the o of the following pairs: $o(\vec{e}, \vec{f})$, $o(\vec{e}, \neg\vec{f})$ (total counts for $\vec{e}$ and any phrases except $\vec{f}$), $o(\neg\vec{e}, \vec{f})$ and $o(\neg\vec{e}, \neg\vec{f})$. Then calculate the $LLR(\vec{e}, \vec{f})$ by using the formula in Figure 2.

2. If the $LLR(\vec{e}, \vec{f})$ exceeds the threshold, the occurrences of $\vec{e}$ and $\vec{f}$ are considered to be reliably correlated. The correlation can be classified as positive if both $ad \quad bc > 0$ in the matrix in Figure 2 and $t(\vec{e}, \vec{f}) > t(\vec{e}, \quad) \cdot t(\quad, \vec{f})$, negative if $ad \quad bc < 0$, and else unreliably correlated.

3. If the LLR value is smaller than the threshold, we cannot make a reliable decision as to whether the occurrences of $\vec{e}$ and $\vec{f}$ are correlated or not.

We produce bilingual concepts only from phrase pairs that are considered to have positive correlation.

$$LLR(\vec{e}, \vec{f}) = 2 \log \frac{B(a|a+b, \frac{a}{a+b}) B(c|c+d, \frac{c}{c+d})}{B(a|a+b, \frac{a+c}{a+b+c+d}) B(c|c+d, \frac{a+c}{a+b+c+d})}$$

$$B(k|n,p) = \binom{n}{k} p^k (1 \quad p)^{n-k}$$

	$\vec{e}$	$\neg\vec{e}$
$\vec{f}$	a	b
$\neg\vec{f}$	c	d

cooccurrence count matrix

Figure 2: Log-Likelihood Ratio Statistics (Dunning, 1993)

3.2.3 Updating Distributions

We update the t- and o-distributions in the same way as the Marcu-Wong method; we calculate the probabilities for each alignment generated during the hillclimbing process over all document pairs in a corpus, and then collect counts over all concepts and coocurrences

in these alignments. The detailed procedure differs from the original as follows because of LLR checking.

***t*-distribution** In the generated alignments, unreliably correlated bilingual concepts are never found because they are suppressed producing by LLR checking. Word sequences that can be generated by such unreliably correlated bilingual concepts are mostly composed of monolingual concepts. Therefore we use the following procedure for updating the t-distribution:

1. For each document pair, collect counts for each concept for all alignments.

2. Distribute the counts for every monolingual concept in the result of step 1 to the every monolingual and unreliably correlated bilingual concepts in proportion to the current t-distribution to obtain smoothed counts for a document pair.

3. Collect these smoothed counts for all document pairs in a corpus.

4. Obtain the updated t-distribution for the next iteration by normalizing the counts.

In our implementation of the suffix array data structure, the difference from the initial distribution is stored in the table for each document pair. Every count for positive and negative correlated bilingual concepts is stored in the table since they cannot be directly calculated from the initial distribution. On the otherhand, the counts for the rest can be obtained by multiplying their initial counts by a factor for each document pair, which is also held in the table.

***o*-distribution** From the definition of phrase cooccurrences described in Section 3.2.1, we approximate the updated cooccurrence counts of $(\vec{e}, \vec{f})$ in $(\boldsymbol{e}, \boldsymbol{f})$ by the following equation ($a, b, \vec{e_i}, \vec{f_j}$ are the same as in Section 3.2.1):

$$\sum_{i=1}^{a}\sum_{j=1}^{b} t(\vec{e_i}, \vec{f_j}|(\boldsymbol{e}, \boldsymbol{f})) + \frac{a + \sum_{n=1}^{b-1}\sum_{c=1}^{a-n} c}{ab} \sum_{i=1}^{a}\sum_{j=1}^{b} t(\vec{e_i}, \ |(\boldsymbol{e}, \boldsymbol{f}))\, t(\ , \vec{f_j}|(\boldsymbol{e}, \boldsymbol{f}))\,. \quad (6)$$

We can easily calculate these conditional probabilities from the difference table for t-distribution if the table also hold the total alignment probability of the document pairs.

4 Experiments

We conducted a series of preliminary experiments using our model to align phrases from the NHK Japanese-English broadcast news corpus, which is composed of document pairs of Japanese news scripts and their manual translation into English. The Japanese documents in the corpus were segmented into morpheme tokens with part-of-speech tags by Chasen[3], the morphological analyzer for Japanese. Each experiment was given different conditions as to the size of corpora, LLR thresholds, and the times of iterations as in Table 1. Note that the smaller corpus is the subset of the larger one.

One human evaluator evaluated the quality of the phrase alignments by marking all alignments from the 10 randomly selected article pairs in each of the above experiments. He marked according to three grades:

correct(○): the extracted phrase pair is parallel without no extra or absent words,

partly correct(△): the extracted phrase pair has extra or absent word(s) but almost all content words are parallel,

incorrect( ): otherwise.

Table 2 shows the number of alignments for each grade, the average number of words in the aligned phrases, and coverage (how many words of each document were covered by the aligned phrases).

Table 3 shows some phrase alignments that have higher LLR scores in the article pair

[3] `http://chasen.naist.jp/hiki/ChaSen/`

No.	Corpus Size # of document pairs (# of tokens / types)	LLR Threshold[4]	Iteration Times
1	1,000 (J: 287,597 / 10,855) (E: 161,976 / 10,521)	3.841 (95%)	1
2			3
3			5
4		2.706 (80%)	
5		0.4549 (50%)	
6	2,000 (J: 578,374 / 18,182) (E: 312,353 / 17,905)	3.841 (95%)	3

Table 1: Experimental Conditions

Condition No.		1	2	3	4	5	6
Evaluation	○	32/7	65/19	102/32	188/59	164/44	173/46
(# of alignments	△	8/4	28/15	35/21	61/46	66/42	53/33
(tokens/types))		42/22	33/19	26/20	216/166	357/258	38/25
rate of ○ or △ (token/type)		.488/.371	.738/.642	.840/.726	.535/.389	.392/.250	.856/.760
Phrase Length	J	1.02	1.09	1.21	1.29	1.23	1.33
(# of words)	E	1.01	1.10	1.19	1.18	1.10	1.24
Coverage	J	.029	.049	.071	.210	.254	.122
(rate in words)	E	.051	.088	.124	.341	.403	.211

Table 2: Results of evaluation

shown in Figure 1 from the experiment for the condition 6.

Comparing the evaluations of the experimental conditions 3 to 5, it is apparent that LLR checking seems to be useful for selecting parallel segments from comparable corpora.

Comparing the conditions 1 to 3, we see that the iteration improves the quality of alignments, but is not very effective for finding new longer alignments as expected. This may be because our method of updating distributions is inappropriate.

Comparing the conditions 3 and 6, we see that a larger corpus size made coverage better and phrase lengths longer but did not change the precision by much. This means that LLR checking guarantees the correctness of phrasal alignments according to the LLR thresholds.

5 Related Work

The studies on acquiring translation knowledge from non-parallel corpora started with extracting lexical translations (e.g. (Fung and Yee, 1998; Rapp, 1999)). To find translations, they generally exploit the tendency that equivalent words have similar contextual words in corpora of different languages. These methods are powerful in terms of their applicability even to unrelated bilingual corpora, but they provide very poor coverage.

Extracting parallel segments of longer than lexical level from non-parallel corpora have been studied afterward. As for the challenges to exploit comparable corpora, there have been some efforts on extracting parallel sentences (Zhao and Vogel, 2002; Munteanu and Marcu, 2002). Both studied used a statistical bilingual dictionary obtained from a parallel corpus as bootstraps to extract more parallel

[4]The asymptotic distribution of LLR statistics will follows $^2(1)$, so if the LLR score of a phrase pair exceeds a threshold whose $^2(1)$ probability is p, the phrase pair is considered to be correlated with an approximate probability of p.

Japanese	English	Log Prob.	LLR	Judge
地震	quake	14.2	12.8	○
地震	earthquakes	15.3	10.1	○
気象庁	The Meteorological Agency	15.9	8.11	○
以上/の	more	15.1	7.89	○
地震	jolted	14.6	7.83	
強い	strong	14.9	4.17	○
を/観測/し (*observe(d)*)	eaethquake	16.8	4.11	

Table 3: Example of phrase alignments extracted in the experiment No.6

sentences and bilingual lexicons from comparable corpora. Fung and Cheung (2004) used a multi-level bootstrapping to improve alignments at the levels of document, sentence, and word pairs and thereby avoid the use of preexisting knowledge sources such as dictionaries.

These methods of parallel sentence extraction have a limitation in that few sentence pairs can be extracted from corpora that are far from parallel. Munteanu and Marcu (2002) proposed a method of extracting sub-sentential parallel fragments from comparable corpora. It first selects sentence pairs which are likely to share some parallel fragments from a bilingual dictionary of broad coverage, then detects parallel fragments within each of the sentence pairs by another precise bilingual dictionary.

These studies aim to *mine* corpora for clean parallel parts in order to acquire further knowledge for proposes such as SMT. On the other hand, our approach directly acquires phrase alignments from comparable document pairs. We obtain lexical translation knowledge and extract parallel parts from comparable corpora simultaneously.

6 Conclusion

We described a method of extracting phrasal alignments from comparable corpora by using an extended phrase-based joint probability model for statistical machine translation. Our method can extract phrasal alignments directly from comparable document pairs composed of about 5–8 sentences without preexisting resources or splitting them into sentences. The experiments showed that our method achieves about 0.8 in precision of phrasal alignment extraction when using 2,000 document pairs of Japanese-English news articles as training data, thanks to its use of the alignment checking process using log-likelihood ratio statistics.

The experiments indicated plenty of room for our method to be improved, e.g.:

> As mentioned before, our method of updating distributions is far from theoretically well-grounded, which may affect performance.
>
> Computation cost is high, especially for the hillclimbing search. We need to make practical improvements to the process (e.g. (Birch et al., 2006)). Calculating distributions on the fly also costs very much, which spoil the merit of the suffix array data structure in part.
>
> Our method cannot recognize discontinuous segments as phrases. It is common that a continuous phrase in English does not have a Japanese counterpart of discontinuous segments because of the difference in language structure. We would like to improve the model so that it can handle discontinuous phrasal segments.
>
> Our method highly depends on the size of each document in a training corpus. Because we find statistical prominence in the cooccurrences distribution to find reliable phrase correspondences, expan-

sion of each cooccurrence window will decrease the performance of our method. We need to test our method for longer documents.

We would like to make a much finer evaluation by manually constructing an evaluation set in the near future. The proposed model highly depends The proposed model is an enhancement of Marcu-Wong's Model 1 and it does not contain a constraint on word or phrase order. We would like to enhance our method by taking order into consideration, and apply it to statistical machine translation.

Acknowledgements

We would like to express our deep gratitude to Mr. Tomoki Kozuru from Kanji Information System Co.,Ltd., who worked together with us in implementing the experimental system.

References

Alexandra Birch, Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings on the HLT-NAACL 2006 Workshop on Statistical Machine Translation*, pages 154–157.

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2005. Scaling phrase-based statistical machine translation to larger corpora and longer phrases. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 255–262.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Pascale Fung and Percy Cheung. 2004. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 1051–1057.

Pascale Fung and Lo Yuen Yee. 1998. An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL '98)*, pages 414–420.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 133–139.

Robert C. Moore. 2004. On log-likelihood-ratios and the significance of rare events. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 333–340.

Dragos Stefan Munteanu and Daniel Marcu. 2002. Processing comparable corpora with bilingual suffix trees. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 289–295.

Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 81–88.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL '98)*, pages 519–526.

Bing Zhao and Stephan Vogel. 2002. Adaptive parallel sentences mining from web bilingual news collection. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2002)*, pages 745–748.

A Greedy Decoder for Phrase-Based Statistical Machine Translation

Philippe Langlais, Alexandre Patry and Fabrizio Gotti
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal,
C.P. 6128, succursale Centre-Ville
H3C 3J7, Montréal, Québec, Canada
{felipe,patryale,gottif}@iro.umontreal.ca

Abstract

Most phrase-based statistical machine translation decoders rely on a dynamic programming technique for maximizing a combination of models, including one or several language models and translation tables. One implication of this choice is the design of a scoring function that can be computed incrementally on partial translations, a restriction a search engine using a complete-state formulation does not have. In this paper, we present experiments we conducted with a simple, yet effective greedy search engine. In particular, we show that when seeded with the translations produced by a state-of-the-art beam search decoder, it produces an output of significantly higher quality than the latter taken alone, as measured by automatic metrics.

1 Introduction

At the beginning of Statistical Machine Translation (SMT), efforts were made to design efficient machine decoders for word-based models (Tillmann et al., 1997; Wang and Waibel, 1997; Niessen et al., 1998; García and Casacuberta, 2001). As phrase-based models gained in popularity (Koehn et al., 2003), specific phrase-based decoders were released, such as Pharaoh[1] (Koehn, 2004) and some open-source alternatives, among which Ramses (Patry et al., 2006) and Phramer (Olteanu et al., 2006).

All these decoders share one common property: they rely on a scoring function that is incremental, in order to allow an efficient organization of the computations by dynamic programming (DP). For the kind of models we typically consider in SMT (word- or phrase-based), this is just fine, but one can easily think of models for which such a property is inappropriate.

One notable exception to the dynamic programming approach is the ReWrite decoder (Germann et al., 2001). It is a greedy decoder that iteratively tries to improve a current translation by modifying some of its elements according to some predefined operations. At each iteration, the best hypothesis found up to that point is kept and used for the next iteration, until convergence is obtained, which typically happens after a few iterations. A time-efficient refinement of this decoder has been described in (Germann, 2003). However, Foster et al. (2003) did report that this decoder produces translations of lower quality than those produced by a DP-decoder.

To our knowledge, there has been no investigation on a greedy decoder designed to maximize the log-linear combination of models traditionally embedded in a phrase-based SMT system. This paper aims at filling this gap.

We show that our implementation, although not as good as a state-of-the-art beam search DP-engine, is not far off. More interestingly, we report experiments on the Europarl corpus where the greedy search algorithm significantly

[1] Moses, available at http://www.statmt.org/moses/ gracefully replaces Pharaoh.

improves the best translation produced by a DP-based decoder. Last, we demonstrate the flexibility of the approach by adding a reversed language model to the set of models consulted to score a translation.

The paper is organized as follows. We first describe our greedy search algorithm in Section 2. The experimental setup as well as the reference beam search DP-engine we used are described in Section 3. In Section 4, we report experiments comparing our greedy implementation with a state-of-the-art phase-based DP-search engine. We conclude our work in Section 5.

2 The greedy search engine

The strategy of `ReWrite`, as described in (Germann et al., 2001) is one of the simplest form of local search algorithms: a hill-climbing search. It uses a complete-state formulation, which means that it searches over the space of possible translations; while a typical beam search DP-decoder will typically explore the space of prefixes of all possible translations. Usually, a local search operates on a single state, which in our case defines the current translation and allows to move to neighboring states according to some predefined operations.

This local search strategy has three interesting characteristics. First, it requires a constant amount of memory, whereas a DP search requires an amount at the very least linear in the source sentence length. Second, it has been reported that local search algorithms indeed often propose a reasonable solution in combinatorial problems (Russell and Norvig, 1995). Third, the function we seek to optimize does not have to evaluate partial translations, a point we develop later on.

On the down side, the greedy search algorithm is obviously not optimal. In some situations, including ours, this is a risk we are willing to take.

The greedy search, which is sketched in Figure 1, depends on the definition of three functions: one that seeds the search with a current state (`seed`), a scoring function (`score`), which takes a candidate translation as an argument and that we seek to maximize, and a function (`neighborhood`), which returns a set of neighboring hypotheses to consider at each iteration.

Require: $source$ a sentence to translate
  $current \leftarrow$ `seed`($source$)
  loop
    $s\_current \leftarrow$ `score`($current$)
    $s \leftarrow s\_current$
    for all $h \in$ `neighborhood`($current$) **do**
      $c \leftarrow$ `score`(h)
      if $c > s$ **then**
        $s \leftarrow c$
        $best \leftarrow h$
    if $s = s\_current$ **then**
      return $current$
    else
      $current \leftarrow best$

Figure 1: Core of the greedy search algorithm.

2.1 The scoring function

In this study, we seek to maximize a log-linear combination of models typically used in state-of-the-art phrase-based DP-engines. In particular, in the first experiments we report, we maximize the very same function that `Pharaoh` maximizes and which is reported in Equation 1:

$$\begin{array}{rlll} Score(e,f) = & \lambda_{lm} & \log p_{lm}(f) & + \\ & \sum_i & \lambda_{tm}^{(i)} \log p_{tm}^{(i)}(f|e) & - \\ & \lambda_w & |f| & - \\ & \lambda_d & p_d(e,f) & \end{array} \quad (1)$$

where the λs are the weighting coefficients, p_{lm} is a language model, p_{tm}^i are different transfer tables (that share the same parameters in our experiments), $|f|$ stands for the length of the translation (counted in words), and $p_d(e,f)$ is a so-called distortion model (we used the simple one described in (Koehn et al., 2003)).

2.2 The neighborhood function

By inspecting translations produced by `Pharaoh`, we defined a set of six operations that can transform a current translation. This is by no means an exhaustive set, and extensions will be considered in future investigations. In particular, we do not yet allow words (or phrases) to be inserted or deleted, two operations that are used by the `ReWrite` decoder (Germann et al., 2001).

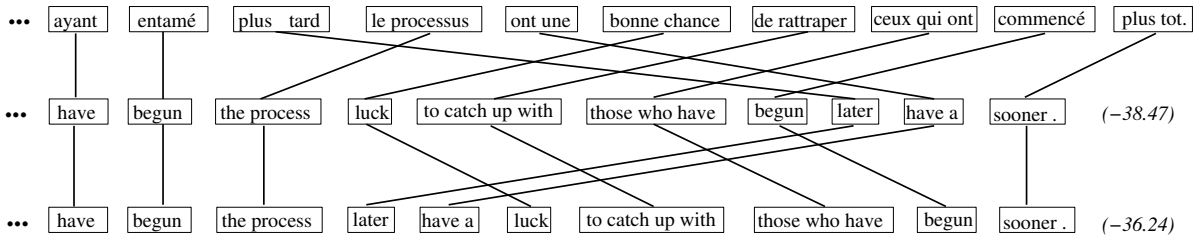


Figure 2: Illustration of an ill-formed translation produced by `Pharaoh` (second line) for an excerpt of a French sentence (first line). The third line shows the translation produced by `feGreedy` after one iteration.

Move The beam search DP-decoder tends to eliminate from the search space hypotheses that cover hard-to-translate segments. Since the decoder is forced to translate all the source material, it is often the case that the translation of those hard-to-translate segments is postponed until the very end of the search, typically producing ill-formed translations (see Figure 2). To overcome this situation to some extent, we allow some target phrases to move within the current translation.

Our implementation is very conservative: whenever two adjacent source phrases are translated by phrases that are distant,[2] we consider moving one of the translation closer to the other.

Swap It happens rather frequently that two adjacent source segments (words or phrases) do not form a phrase that belongs to the transfer table. The order in which their respective translations will be output will be strongly influenced by the compromise between the possible inversions the language model allows and the strong bias toward monotonous translations the distortion model has. For this reason, we defined an operation which allows to swap two adjacent target segments. The complexity of this operation[3] is $\mathcal{O}(N-1)$, that is, linear in the number N of source phrases in the current hypothesis.

Replace This operation simply allows to change the translation given for a specific source segment by another one found in the transfer table. This operation has a complexity of $\mathcal{O}(N \times T)$, where T is the maximum number of translations considered per source phrase.[4]

Bi-replace With the same idea in mind, we allow the translation of two adjacent source phrases to change simultaneously. We hope that by changing more than one unit, the search will likely escape a local maximum. The complexity of this operation is $\mathcal{O}(T^2 \times (N-1))$, that is, quadratic in the number of translations considered per source phrase.

Split One task a beam search DP-decoder handles—most of the time implicitly—is the segmentation of the source material into phrases. We allow our decoder to split in two parts a given source phrase. While doing so, the two new source phrases receive a translation found in the transfer table (we consider all of them). The complexity of this operation is $\mathcal{O}(N \times S \times T^2)$, where S is the (average) number of words a source phrase has in the current hypothesis.

Merge As opposed to the `split` operation, the merge operation allows two adjacent source phrases to be merged, in which case a new translation is also picked from the translation table. This operation is $\mathcal{O}(T \times (N-1))$.

2.3 The seed translation

2.3.1 From scratch

In `ReWrite`, the seed translation is formed by collecting for each word its best translation as provided by the transfer table. This is the idea we implemented as well. There is one subtlety however, when we deal with phrases: a segmentation of the source sentence S into phrases must

[2] As defined by a threshold value counted in words. We used 3 in our experiments.

[3] We measure complexity here in terms of the maximum number of hypotheses that will be considered, given a current one.

[4] A typical value of T in our experiments is 10.

be performed. Since many source phrases overlap, there are many a priori segmentations we can choose from. In our case, we select the segmentation which involves the minimum number of source phrases belonging to the translation model $\mathcal{M}$ that cover maximally the source sentence S.

To do so, it suffices to consider $\mathcal{M}$ as a set of *spans* $\langle i, j \rangle$ denoting the fact that a source phrase in $\mathcal{M}$ covers the positions i to j (counted in words) in S. We define an item τ_s as a triple $\langle b, c, n \rangle$ which respectively stores b, the beginning of a span $\langle b, s \rangle$ ending in position s; c, the number of source words covered so far, and n, the number of source phrases used to cover S up to position s. Intuitively, an item τ_s stores the best coverage found so far from the beginning of the source sentence to position s, along with the number of source phrases used so far.

We compute the item $\tau_{|S|}$ by the recursion described in Equation 2, where we define for an item $\tau \equiv \langle b, c, n \rangle$, the operators $b(\tau)$, $c(\tau)$ and $n(\tau)$ to be respectively b, c and n.

$$\tau_s = \max \left\{ \begin{array}{l} \langle 0, 0, 0 \rangle, \\ \max\limits_{\substack{d \leq s \ : \\ \langle d, s \rangle \in \mathcal{M}}} \left\{ \begin{array}{l} \langle d, \\ c(\tau_d) + s - d + 1, \\ n(\tau_d) + 1 \qquad \rangle \end{array} \right\} \end{array} \right\} \quad (2)$$

The maximizations involved in Equation 2 are carried out over a set of items. We use the following operator to compare two items:

$$\max\{\tau_1, \tau_2\} = \left\{ \begin{array}{ll} \tau_2 & \text{if} \left\{ \begin{array}{ll} c(\tau_1) < c(\tau_2) & \text{or} \\ c(\tau_1) = c(\tau_2) & \text{and } n(\tau_1) > n(\tau_2) \end{array} \right. \\ \tau_1 & \text{otherwise} \end{array} \right. \quad (3)$$

The coverage is obtained by simply backtracking from item $\tau_{|S|}$, that is, by computing the set $\beta\tau_{|S|}$:

$$\beta_{\tau_e} = \left\{ \begin{array}{ll} \phi & \text{if } e \equiv 0 \\ \{\langle b(\tau_e), e \rangle\} \cup \beta(\tau_{\delta(e)}) & \text{otherwise} \end{array} \right. \quad (4)$$
$$\text{with } \delta(e) = \text{argmax}_{b<e}\ c(\tau_b) \neq 0$$

The recursion involved in this computation lends itself to an efficient computation by dynamic programming. Once the source segmentation is found, we simply pick for each source phrase the best translation found in $\mathcal{M}$. An illustration of the segmentation obtained for a short source sentence is provided in Figure 3.

2.3.2 Seeding `feGreedy` with `Pharaoh`

It is likely that a DP-search will outperform our greedy implementation, hereafter named `feGreedy`. Therefore, it is natural to investigate whether any benefit would result from seeding `feGreedy` with the best translation produced by `Pharaoh`.[5]

The idea of cascading two translation engines has been pioneered within the word-based Candide translation system (Berger et al., 1994). Unfortunately, the authors did not describe their local search engine, neither did they provide an evaluation of its benefits to the overall system. The cascading strategy received a more dedicated treatment in Marcu (2001) and Watanabe and Sumita (2003). In their work, the authors were seeding a word-based greedy search algorithm with examples extracted from a translation memory hoping to bias the search toward a better solution. Our motivation is slightly different however: we simply want to know whether the greedy strategy can overcome some search errors made by a phrase-based DP-search.

3 Experimental setup

3.1 Corpora

We concentrated our efforts on the shared task of last year's workshop on Statistical Machine Translation (Koehn and Monz, 2006) which consisted in translating Spanish, German and French texts into English and the reverse direction. The training material available is coming from the Europarl corpus. Four disjoint corpora were released during this exercise, namely `train`, a portion of 688,031, 730,740 and 751,088 pairs of sentences for French, Spanish and German respectively; `dev`, a development corpus that we used for tuning; `devtest`, a corpus of 2,000 pairs of

[5] We used the `--trace` option of `Pharaoh` to access the phrasal alignment corresponding to the best translation found.

F	de plus , nos systèmes administratifs doivent être modernisés . nous devons également donner le bon exemple .
E	in addition , our administrative systems must be modernised , and it is our duty to lead by example .
S_0	[de plus ,] [nos systèmes administratifs] [doivent] [être modernisés] [. nous devons également] [donner le bon exemple .]
T_0	[furthermore ,] [our administrative systems] [must] [modernization] [and we also need] [set a good example .] *-19.5068*
S_1	[de plus ,] [nos systèmes administratifs] [doivent] [être modernisés] **[.] [nous devons également]** [donner le bon exemple .]
T_1	[furthermore ,] [our administrative systems] [must] [modernization] **[.] [we must also]** [set a good example .] SPLIT *-17.4382*
S_2	[de plus ,] [nos systèmes administratifs] [doivent] **[être] [modernisés]** [.] [nous devons également] [donner le bon exemple .]
T_2	[furthermore ,] [our administrative systems] [must] **[be] [modernized]** [.] [we must also] [set a good example .] SPLIT *-15.8488*
S_3	[de plus ,] [nos systèmes administratifs] [doivent] [être] [modernisés] [.] [nous devons également] **[donner] [le bon exemple .]**
T_3	[furthermore ,] [our administrative systems] [must] [be] [modernized] [.] [we must also] **[give] [a good example .]** SPLIT *-15.5885*
S_4	**[de plus ,]** [nos systèmes administratifs] [doivent] [être] [modernisés] [.] [nous devons également] [donner] [le bon exemple .]
T_4	**[in addition ,]** [our administrative systems] [must] [be] [modernized] [.] [we must also] [give] [a good example .] REPLACE *-15.5199*

Figure 3: Steps involved by the translation of a French sentence (F); E is its reference translation. A segmentation (S_0) is first chosen from the 49 different source phrases that cover partially F. T_0 is the associated seed translation. The phrases in bold are those involved in the highest-scored operation at each iteration. Over 4,100 hypotheses have been evaluated within a time period of 300 milliseconds.

sentences that we used for monitoring our system; and `test`, the official test set of the 2006 shared task, that we used only for final tests. We further split the `test` corpus in two parts, `test-in`, the in-domain part which consists of 2,000 sentences from the European parliament debates, and `test-out`, which counts 1,034 sentences[6] collected from editorials of the Project Syndicate website.

3.2 Phrase-based engine

The reference system we used for comparison purposes is the state-of-the-art phrase-based engine which was made available by the organizers of the shared task. The language model (a trigram) was trained using the SRILM toolkit (Stolcke, 2002), and the translation tables (phrases up to 7 words long) were obtained by running the scripts provided. These tables contain 4 scores (relative frequencies and lexical scores in both direction) that each receives a weighting coefficient. A fifth score is intended to serve as a phrase penalty model. The `Pharaoh` built-in distortion model and a word penalty component receive as well a weighting coefficient. Altogether, 8 coefficients were tuned using the script `minimum-error-rate-training.perl`.

For most of our experiments, the threshold values that `Pharaoh` uses were left to their built-in defaults. This is the version of our DP-system that we call BASE from now on.

[6] We removed 30 sentences with encoding problems.

Systems	L	en→L WER	en→L BLEU	L→en WER	L→en BLEU
BASE	fr	55.12	30.16	51.47	29.23
G-S	fr	57.38	24.23	53.99	24.52
G-BASE	fr	**53.62**	**30.64**	**50.37**	**29.62**
BASE	es	55.04	28.17	50.97	29.94
G-S	es	56.86	22.77	53.66	24.80
G-BASE	es	**53.14**	**28.72**	**50.04**	**30.30**
BASE	de	62.38	17.32	60.12	24.54
G-S	de	66.13	13.34	59.90	19.23
G-BASE	de	**61.85**	**17.51**	**58.33**	24.97

Table 1: Performances of different search algorithms measured on the `devtest` corpus, as a function of the translation direction. The figures in bold are significantly better than the corresponding BASE configuration at the 99% confidence level.

4 Experiments

4.1 `feGreedy` with or without `Pharaoh`

We first compared `feGreedy` with BASE by running both decoders, with the same function to maximize (see Equation 1). In one version of the greedy search, G-S, the search was initiated from scratch (see Section 2.3.1). In a second version, G-BASE, the search was seeded with the best translation produced by `Pharaoh` (see Section 2.3.2). The results are reported in Table 1.

Expectedly, for all translation directions, the greedy search algorithm alone provides translations of significantly lower quality than the DP-search. This is consistent with the observations made by (Foster et al., 2003) in word-based translation experiments. However, we observe that the greedy search improves upon the best translation that BASE found. This seems to be consistent for all translation directions and for both evaluation metrics considered. For all translation directions except German-to-English, the improvements are significant at the 99% confidence level.[7]

In order to better appreciate the situation, we report in Table 2 more specific information on what the greedy search accomplishes. We restrict ourselves to translating into English, since this corresponds to the most studied translation direction in the SMT literature, and we did not notice clear differences in the reverse direction.

First of all, we observe that roughly 40% of the translations produced by BASE get improved in score (Equation1) by `feGreedy`. We were expecting a much lower improvement proportion. One explanation for that might be the stack-size limit `Pharaoh` considers as a default (100). Keeping the first hundred best hypotheses for each source coverage (i.e. the number of source words covered by a given hypothesis) might bias the search toward locally optimal hypotheses. More expectedly, however, we observe that more than 90% of the seed translations computed by the technique described in Section 2.3.1 get improved by `feGreedy`.

Regarding the selected operations at each iteration, roughly 40% of them are replacement ones, that is, the replacement of one translation by another one. The `move` operation also highly beneficial. The fact that more than 15% of the winning operations in G-BASE are split operations might appear surprising at first. Recall that this operation comes along with a possible change in the target material and is therefore not just a matter of segmenting differently the source material. We also observe that some operations are only marginally useful. This is the case of `merge` and `swap`. The fact that the `swap` operation is not productive just indicates that the phrase table is already doing a good job at capturing local word-order differences. We do not have yet a clear explanation for the low impact of the `merge` operation.

Last, we can see from Table 2 that the distribution of the number of iterations required by G-BASE and G-S are very different. The former configuration requires only a few iterations to converge: at most 2 iterations in approximatively 70% of the cases. For the latter, only more than half of the translations are completed after 4 iterations. Both versions require less than 10 iterations on average to produce a translation.

It is worthwhile to note that, although we did not yet pay attention to translation speed within our current implementation,[8] `feGreedy` defi-

[7] In all our experiments, we used the bootstrap resampling method described in (Zhang and Vogel, 2004) to compute significance levels, evaluating 1,000 samplings of 700 sentences each.

[8] It is a simple matter to improve the speed of

	fr→en		es→en		de→en	
	G-B	G-S	G-B	G-S	G-B	G-S
%up	42.6	93.5	37.1	90.8	42	95.8
↑ log-s	3.6	2.9	2.7	1.7	1.8	2.9
$\%it. < 2$	44.6	13.5	50.7	13.8	43.1	6.5
$\%it. < 3$	66.2	29.7	74.4	31.6	65.7	17.2
$\%it. < 5$	90.8	59.7	93.3	65.7	91.7	45.0
$\%it. < 10$	98.8	95.0	100.0	97.8	100.0	87.5
MOVE	42.2	–	44.0	–	42.1	–
REPLACE	41.3	45.1	38.3	45.3	37.7	51.7
SPLIT	14.9	52.8	16.3	52.4	18.6	46.5
MERGE	0.9	1.7	0.8	1.8	1.0	1.1
SWAP	0.5	0.2	0.2	0.2	0.3	0.5

Table 2: Profile of two variants of `feGreedy` on the `devtest` corpus. G-B is a shorthand for G-BASE. %up stands for the percentage of sentences that get improved by the greedy search. ↑ log-s indicates the average gain in score (Equation 1). $it. < n$ indicates the percentage of sentences improved for which less than n iterations were required. The bottom part of the table indicates the percentage of operations that ranked best at an iteration of the greedy search.

nitely compares favorably to BASE in that respect. Currently, translating the 1,000 sentences of `devtest` on a Pentium computer clocked at 3 GHz requires 9 minutes with `feGreedy`, compared to 78 minutes with BASE.

4.2 Further experimenting with `feGreedy`

In the previous section, we conducted a pairwise comparison of `feGreedy` with our reference system, by providing the greedy decoder the same function `Pharaoh` is maximizing. In this section, we report experiments we conducted in order to improve `feGreedy`. Our starting point is the configuration of the greedy search seeded with the best translation produced by BASE.

4.2.1 Adding new features

One strength of the greedy search is that it operates on a full candidate translation. This allows us to optimize a scoring function which is not necessarily incremental. To illustrate this added flexibility, we added a reversed n-gram language model to the set of models of the scoring function maximized by `Pharaoh`. We call this variant G-REV.

A reversed n-gram model simply predicts each word of the translation from right to left, as described in Equation 5. At first glance, this might seem like an odd thing to do, since there is probably not much information a decoder can gain from this model. Yet, this is one of the simplest models imaginable, which could not be easily integrated into a DP-decoder such as `Pharaoh`, since the suffix of a hypothesis is unknown during the search.

$$p(t_1^T) \approx \prod_{i=1}^{T} p(t_i | t_{i+1} \dots t_{i+n-1}) \qquad (5)$$

Because we added a new model to the linear combination optimized by `feGreedy`, we had to tune the coefficients involved once more. To save some computation time, however, we did not explore the full range of values for each coefficient, but concentrated on values close enough to those we had already found. The results of this experiment are reported in Table 3.

For all translation directions but Spanish-to-English, the gain in performance, as measured by WER, are very small if not negative. However, improvements in BLEU, although mostly not significant, are consistent for all translation directions.

		en→L		L→en	
Systems	L	WER	BLEU	WER	BLEU
BASE	fr	55.12	30.16	51.47	29.23
G-BASE	fr	53.62	30.64	50.37	29.62
G-REV	fr	53.65	30.85	50.30	29.70
BASE	es	55.04	28.17	50.97	29.94
G-BASE	es	53.14	28.72	50.04	30.30
G-REV	es	52.37	29.31	50.05	30.33
BASE	de	62.38	17.32	60.12	24.54
G-BASE	de	61.85	17.51	58.33	24.97
G-REV	de	61.85	17.57	57.99	25.20

Table 3: Performances of the G-REV variant for different translation directions, measured on the `devtest` corpus.

`feGreedy`, since in our current implementation, any operation applied to a hypothesis triggers the computation of its score from scratch, while some straightforward bookkeeping would eliminate most of the computations.

4.2.2 A beam-search version of `feGreedy`

As we already noted, one advantage of the greedy search is that it requires a set amount of memory, since it does not build a search graph like DP-search engines do (e.g. `Pharaoh`). This is an interesting advantage, but keeping only a single-best current translation is somehow too heavy-handed a response to the memory problem. Therefore, in this experiment, we tested a variant of the greedy search, technically known as local beam search (Russell and Norvig, 1995). In this greedy search, a beam of at most k best hypotheses are kept at each iteration. The search tries to improve on each of them, until no improvement can be found. We call this version G-BEAM.

We populate the beam with k seed hypotheses. One is the best translation proposed by BASE, as described in section 2.3.2. The $k-1$ others are derived from the source coverage we compute, as described in Section 2.3.1. To form the ith seed translation, we select the ith-best translation of each source phrase, as found in the transfer table. Obviously, they are many other ways we could proceed to produce k seed translations, including considering the k-first hypotheses produced by BASE. An example of seed translations produced for one short sentence is reported in Figure 4. In this example, as is often the case, the seed hypothesis proposed by BASE is ranked higher than the one computed from scratch.

No improvement in BLEU and WER have been observed over the 1-best greedy search seeded with `Pharaoh` (G-BASE). This is disappointing, but not entirely surprising, since BASE already does a good job, and that G-BASE further improved on it. What is more interesting, however, is that the beam version of our greedy search managed to find higher-scored translations (according to Equation 1) than G-BASE does. On one hand, this is satisfactory from a search point of view. On the other hand, it is disturbing to note that search errors are at some point beneficial! The adequacy of the evaluation metrics we considered might be one reason for this observation. However, we believe that the problem is more likely due to severe (well-known) shortcomings of the scoring function we seek to maximize, including its blindness to syntactical quality.

Averaged across all translation directions,

cette question est , bien sûr , parfaitement légitime , mais il faut y répondre de façon correcte et précise . *(source sentence)*
◇ this question is , of course , perfectly legitimate , but it must be answered properly and carefully. (`Pharaoh`, *-16.11*)
◇ subject is of course , perfectly legitimate , but we must respond to properly and carefully. (scratch-1, *-18.22*)
◇ subject is of course fully justified , but it must be answered properly and carefully. (scratch-3, *-20.58*)
◇ subject is of course perfectly quite legitimate , but it must be answered properly and carefully. (scratch-2, *-21.57*)

Figure 4: 4 seed translations computed for the source (French) sentence at the top along with their score. *scratch-n* stands for a seed translation computed from scratch, picking for each source phrase belonging to the coverage, the nth translation found in the transfer table.

roughly 20% of the translations produced by G-BEAM are different from those produced by G-BASE. Among these modified translations, 87% have a higher score (Equation 1). The fact that for some sentences, G-BEAM missed an optimum found by G-BASE is simply due to the greediness of the search along with a limited beam size. We observed that by increasing the beam size, the number of downgraded translations produced by G-BEAM decreases. By simply choosing the best-scored translation produced by either G-BASE or G-BEAM, we did not manage to improve significantly BLEU and WER figures.

4.2.3 Final tests

We conclude our exploration of `feGreedy` by running on the `test` corpus the most salient versions we tested on the development corpus: BASE, the `Pharaoh` DP-decoder, G-BASE, the greedy search engine seeded with the best translation BASE found, G-BEAM-5, the local beam variant of `feGreedy`, with a beam size of 5, and G-REV, the greedy variant using a reversed language model.

Results are reported in Table 4 and 5 for the in- and out-domain test material respectively.

Systems	L	en→L WER	en→L BLEU	L→en WER	L→en BLEU
BASE	fr	54.85	30.90	51.69	29.96
G-BASE	fr	**53.38**	**31.42**	**50.46**	**30.27**
G-BEAM-5	fr	**53.46**	31.26	**50.40**	30.13
G+B5	fr	**53.43**	**31.28**	**50.36**	30.17
G-REV	fr	**53.49**	**31.52**	**50.48**	**30.25**
BASE	es	54.23	29.64	51.04	30.54
G-BASE	es	**52.77**	**30.14**	**50.02**	**30.87**
G-BEAM-5	es	**52.61**	30.24	**50.12**	**30.89**
G+B5	es	**52.61**	**30.25**	**50.11**	**30.93**
G-REV	es	**52.67**	29.79	**50.07**	30.84
BASE	de	62.32	17.68	60.54	24.45
G-BASE	de	**61.73**	17.88	**58.85**	24.66
G-BEAM-5	de	61.98	17.82	**57.62**	24.59
G+B5	de	61.95	17.84	**57.62**	24.58
G-REV	de	**61.77**	17.89	**58.48**	24.82

Table 4: Performances of different search algorithms measured on the `test-in` corpus. Figures in bold are significantly better than their BASE counterpart at the 99% confidence level.

Systems	L	en→L WER	en→L BLEU	L→en WER	L→en BLEU
BASE	fr	60.29	22.31	56.66	20.78
G-BASE	fr	**57.80**	**23.44**	**54.70**	**21.38**
G-BEAM-5	fr	**57.68**	**22.91**	**54.44**	**21.28**
G+B5	fr	**57.61**	**23.03**	**54.43**	**21.33**
G-REV	fr	**58.12**	**23.25**	**54.66**	**21.37**
BASE	es	57.07	24.20	51.11	25.17
G-BASE	es	**54.83**	**25.09**	**49.77**	**25.59**
G-BEAM-5	es	**54.16**	24.91	**49.74**	**25.74**
G+B5	es	**54.11**	**24.95**	**49.72**	**25.69**
G-REV	es	**53.46**	**26.33**	**49.80**	**25.64**
BASE	de	67.09	11.00	65.62	16.00
G-BASE	de	**65.79**	**11.49**	**63.51**	16.38
G-BEAM-5	de	**66.12**	11.24	**61.54**	**16.72**
G+B5	de	**66.10**	11.33	**61.53**	**16.74**
G-REV	de	**65.93**	11.40	**62.96**	16.38

Table 5: Performances of different search algorithms measured on the `test-out` corpus. Figures in bold are significantly better than their BASE counterpart at the 99% confidence level.

First, we observe that the greedy variant G-BASE outperforms the BASE algorithm, for both in- and out-domain. The improvements in WER and BLEU are significant (at the 99% confidence level) for all translation directions, but German-to-English. This is consistent with our previous experiments on the development corpus.

Second, the beam version of `feGreedy`, although significantly better than BASE in most cases, performs usually marginally worse than the corresponding G-BASE variant. The observation we made on the development corpus still holds: the beam variant of the search manages to find translations that are better scored by Equation 1. On the out-domain (resp. in-domain) corpus, 34% (resp. 17%) of the translations produced by G-BEAM-5 did improve in score compared with their G-BASE counterpart. Less than 4% (resp. 3%) received a lower score. The fact that, on the out-domain corpus, twice as many translations receive an higher score with the beam version is encouraging, even if it does not clearly pay off in terms of evaluation metrics.

Picking the highest-scored translation (Equation 1) proposed by either G-BASE or G-BEAM-5 slightly improves upon the G-BEAM-5 variant for almost all translation directions, but the gain is not significant. The corresponding figures are reported as the G+B5 variant in Tables 4 and 5.

5 Conclusions

In this study, we addressed the problem of searching the space of possible translations with a greedy search algorithm designed to maximize the log-linear function many state-of-the-art phrase-based systems use. We discussed some advantages of search algorithms working on a complete-state representation as our greedy search does. We conducted experiments showing that it could improve the best translation found by the more demanding multi-stack beam-search dynamic-programming algorithm embedded in decoders such as `Pharaoh` or `Ramses`.

Perhaps the main contribution of this study is to point out the potential such an easy search algorithm has over more demanding decoders. Until now, this was an idea that had not received much attention in the phrase-based SMT community.

We plan to extend this work in several directions. Actually, one initial motivation for this

study was to explore post-processing operations that could apply to the output of a translation engine, in order to recover systematic errors, in a way inspired by transformation-based learning (Brill, 1995). On step toward accomplishing this consists in increasing the number of operations that our greedy search can perform, associating with each of them a coefficient that we can adjust on a development corpus. This is the idea we want to explore further.

We also want to cast our greedy decoder within the open-source framework called `Mood`, whose principle is to offer decoders that are easy to modify and extend. Therefore, our goal will be to release a reengineered version of `feGreedy`.

6 Acknowledgements

This study has been partially funded by a NSERC grant. We are grateful to Pierre Poulin for his fruitful comments on this work.

References

Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, John R. Gillett, John D. Lafferty, Robert L. Mercer, Harry Printz, and Luboš Ureš. 1994. The Candide system for machine translation. In *Proceedings of HLT*, pages 157–162, Morristown, NJ, USA.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

George Foster, Simona Gandrabur, Philippe Langlais, Pierre Plamondon, Graham Russel, and Michel Simard. 2003. Statistical machine translation: Rapid development with limited resources. In *MT Summit IX*, pages 110–117, New Orleans.

Ismael García and Francisco Casacuberta. 2001. Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *Proceedings of the 8th MT Summit*, pages 115–120, Santiago de Compostela, Spain.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2001. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting of the ACL*, pages 228–235, Toulouse, France.

Ulrich Germann. 2003. Greedy decoding for statistical machine translation in almost linear time. In *HLT-NAACL*, pages 72–79, Edmonton, Canada.

Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between European languages. In *Proceedings of the ACL Workshop on Statistical Machine Translation*, pages 102–121, New York City, June.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT*, pages 127–133.

Philipp Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based SMT. In *Proc. of the 6th AMTA*, pages 115–124, Washington, DC.

Daniel Marcu. 2001. Towards a unified approach to memory- and statistical-based machine translation. In *Proceedings of the 39th Annual Meeting of the ACL*, pages 378–385, Toulouse, France.

Sonia Niessen, Stephen Vogel, Hermann Ney, and Christof Tillmann. 1998. A DP-based search algorithm for statistical machine translation. In *Proceedings of the 36th Annual Meeting of the ACL and 17th COLING*, pages 960–966, Montréal, Canada.

Marian Olteanu, Chris Davis, Ionut Volosen, and Dan Moldovan. 2006. Phramer – an open source statistical phrased-based translator. In *Proceedings of the HLT/NAACL Workshop on Statistical Machine Translation*, pages 150–153, New York, USA.

Alexandre Patry, Fabrizio Gotti, and Philippe Langlais. 2006. Mood: A modular object-oriented decoder for statistical machine translation. In *5th LREC*, pages 709–714, Genoa, Italy, May.

Stuart Russell and Peter Norvig. 1995. *Artificial Intelligence. A Modern Approach*. Prentice Hall.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of ICSLP*, Denver, Colorado, Sept.

Christoph Tillmann, Stephen Vogel, Hermann Ney, and A. Zubiaga. 1997. A DP-based search using monotone alignments in statistical translation. In *Proceedings of the 35th Annual Meeting of the ACL*, pages 289–296, Madrid, Spain.

Ye-Yi Wang and Alex Waibel. 1997. Decoding algorithm in statistical machine translation. In *Proceedings of the 35th Annual Meeting of the ACL*, pages 366–372, Madrid, Spain.

Taro Watanabe and Eiichiro Sumita. 2003. Example-based decoding for statistical machine translation. In *Machine Translation Summit IX*, pages 410–417, New Orleans, Louisiana.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of the 10th TMI*, pages 85–94, Baltimore, Maryland, USA.

Alignment-Guided Chunking

Yanjun Ma, Nicolas Stroppa, Andy Way
National Centre for Language Technology
School of Computing
Dublin City University
Glasnevin, Dublin 9, Ireland
{yma,nstroppa,away}@computing.dcu.ie

Abstract

We introduce an adaptable monolingual chunking approach–Alignment-Guided Chunking (AGC)–which makes use of knowledge of word alignments acquired from bilingual corpora. Our approach is motivated by the observation that a sentence should be chunked differently depending the foreseen end-tasks. For example, given the different requirements of translation into (say) French and German, it is inappropriate to chunk up an English string in exactly the same way as preparation for translation into one or other of these languages.

We test our chunking approach on two language pairs: French–English and German–English, where these two bilingual corpora share the same English sentences. Two chunkers trained on French–English (*FE-Chunker*) and German–English (*DE-Chunker*) respectively are used to perform chunking on the same English sentences. We construct two test sets, each suitable for French–English and German–English respectively. The performance of the two chunkers is evaluated on the appropriate test set and with one reference translation only, we report F-scores of 32.63% for the *FE-Chunker* and 40.41% for the *DE-Chunker*.

1 Introduction

Chunking plays an important role in parsing, information extraction and information retrieval. Chunking is often a useful preprocessing step for many bilingual tasks, such as machine translation, cross language information retrieval, etc.

We introduce an adaptable chunking approach guided by word alignments automatically acquired from a bilingual corpus. Our approach is motivated by the observation that a sentence should be chunked differently depending the end-task in mind. Our approach employs bilingual word alignment in training and is tested on the monolingual chunking task. Our goal is to build adaptable monolingual chunkers for different language pairs, with the aim of facilitating bilingual language processing tasks.

We investigate our chunking approach on two language pairs: French–English and German–English, where these two bilingual corpora share the same English sentences. Two chunkers trained on French–English (*FE-Chunker*) and German–English (*DE-Chunker*) respectively are used to perform chunking on the same English sentences. We construct two test sets, each suitable for French–English and German–English respectively. The performance of the two chunkers is evaluated on the appropriate test set and with one reference translation only, we report F-scores of 32.63% for the *FE-Chunker*

and 40.41% for the *DE-Chunker*. We also extend our chunking approach with *Multi-level Chunking*, which is more tolerant of any chunking errors obtained.

The remainder of this paper is organized as follows. In Section 2, we review the previous research on chunking including monolingual chunking and bilingual chunking. Section 3 describes our chunking method. In Section 4, the experimental setting is described. In Section 5, we evaluate our chunking method on a one-reference 'gold standard' testset. Section 6 concludes the paper and gives avenues for future work.

2 Previous Research

2.1 Monolingual Chunking

Most state-of-the-art monolingual chunking methods are linguistically motivated. The CoNLL-2000 shared task (Tjong Kim Sang and Buchholz, 2000) defined chunking as dividing text into syntactically related non-overlapping groups of words. Chunks are directly converted from the Penn Treebank (Marcus et al., 1993) and each chunk is labelled with a specific grammatical category, such as NP, VP, PP, ADJP etc. This chunking method is sensitive to the grammars of a specific language and performs chunking in a monolingual context.

Marker-based chunking is another syntax-aware chunking strategy. This chunking approach is based on the "Marker Hypothesis" (Green, 1979), a psycholinguistic constraint which posits that all languages are marked for surface syntax by a specific closed set of lexemes or morphemes which signify context. Using a set of closed-class (or "marker") words, such as determiners, conjunctions, prepositions, possessive and personal pronouns, aligned source–target sentences are segmented into chunks. A chunk is created at each new occurrence of a marker word, with the restriction that each chunk must contain at least one content (or non-marker) word. Although marker-based chunking has been used in bilingual tasks such as machine translation between European languages (Gough and Way, 2004; Groves and Way, 2005; Stroppa and Way, 2006), which are relatively similar with regard to marker words and word orders, it is less appropriate for language pairs as different as Chinese and English (Ma, 2006).

2.2 Bilingual Chunking

Bilingual chunkers are usually based on parsing technology. (Wu, 1997) proposed Inversion Transduction Grammar (ITG) as suitable for the task of bilingual parsing. The stochastic ITG brings bilingual constraints to many corpus analysis tasks such as segmentation, bracketing, and parsing, which are usually carried out in a monolingual context. However, it is difficult to write a broad bilingual ITG grammar capable of dealing with long sentences. (Wang et al., 2002) proposed an algorithm integrating chunking and alignment and obtained good precision. However, this method needs quite a lot of syntax information and prior knowledge. (Liu et al., 2004) proposed an integrated probabilistic model for bilingual chunking and alignment independent of syntax information and grammatical rules.

3 Alignment-Guided Chunking

3.1 Notation

While in this paper, we focus on both French–English and German–English, the method proposed is applicable to any language pair. The notation however assumes the French–English task in what follows.

Given a French sentence f_1^I consisting of I words $\{f_1, \ldots, f_I\}$ and an English sentence e_1^J consisting of J words $\{e_1, \ldots, e_J\}$, $A_{F \to E}$ (resp. $A_{E \to F}$) will denote a French-to-English (resp. an English-to-French) word alignment between f_1^I and e_1^J. As 1-to-n alignments are quite common, $A_{F \to E}$ can be represented as a set of pairs $a_i = \langle f_i, E_i \rangle$ denoting a link between one single French word f_i and a few English words E_i (and similarly for $A_{E \to F}$). The set E_i is empty if the word f_i is not aligned to any word in e_1^J.

Given a French–English sentence pair $\langle f_1^I, e_1^J \rangle$, suppose f_i is aligned to a set of En-

glish words $E_i = \{e_j, \ldots, e_{j+m}\}$, and $E_{i+1}^I = E_{i+1} \bigcup \cdots \bigcup E_I = \{e_k, \ldots, e_{k+n}\}$ denotes a union of English words that are aligned to the set of French words $\{f_{i+1} \ldots, f_I\}$. There should be a partition between f_i and f_{i+1}, iff. $k > j + m$. We can partition the English sentence using the same method.

Given a French–English sentence pair and the word alignment between them, we can partition both French and English sentences following the criteria described above. As this chunking is guided by the word alignment, we call it *Alignment-Guided Chunking*.

Assume the French–English sentence pair and their word alignment in (1):

(1) *French*: Cette ville est chargée de symboles puissants pour les trois religions monothéistes .

English: The city bears the weight of powerful symbols for all three monotheistic religions .

Word alignment: 0-0 1-1 2-2 3-4 4-5 5-7 6-6 7-8 8-9 9-10 10-12 11-11 12-13

The AGC chunks derivable via our method are displayed in Figure 1.

Cette ||| ville ||| est ||| chargée ||| de ||| symboles puissants ||| pour ||| les ||| trois ||| religions monothéistes ||| .

The ||| city ||| bears ||| the weight ||| of ||| powerful symbols ||| for ||| all ||| three ||| monotheistic religions ||| .

Figure 1: Example of AGC chunks

Note that the method is able to capture adjective–noun combinations in each language, as well as the determiner-noun pair in English.

3.2 Data Representation

(Ramshaw and Marcus, 1995) introduced a data representation for baseNP chunking by converting it into a tagging task: words inside a baseNP were marked I, words outside a baseNP receive an O tag, and a special tag B was used for the first word inside a baseNP immediately following another baseNP. (Tjong Kim Sang and Veenstra, 1999) examined seven different data representations for noun phrase chunking and showed that the choice of data representation has only a minor influence on chunking performance.

In our chunking approach, every word is classified into a chunk and no fragments are left in a sentence. Accordingly, we do not need the tag O to mark any word outside a chunk. We can employ three data representations similar to (Tjong Kim Sang and Veenstra, 1999) named IB, IE, IBE1, IBE2, where the I tag is used for words inside a chunk. They differ in their treatment of chunk-initial and chunk-final words as shown in Table 1.

In our experiments, we use IE to represent the data, so that the problem of chunking is transformed instead into a binary classification task. The IE tag representation for the English sentence in Figure 1 is shown in (2):

(2) The/E city/E bears/E the/I weight/E of/E powerful/I symbols/E for/E all/E three/E monotheistic/I religions/E ./

Again, note the dependence of determiners and adjectives on their following head noun.

3.3 Parameter Estimation

In this section, we briefly introduce two well-known machine learning techniques we used for parameter estimation, namely Maximum Entropy (MaxEnt) and Memory-based learning (MBL). Both of them are widely used in Natural Language Processing (NLP).

Maximum Entropy was first introduced in NLP by (Berger et al., 1996). It is also used for chunking (Koeling, 2000). Memory-based learning (e.g. (Daelemans and Van den Bosch, 2005)) is based on the simple twin ideas that:

- learning is based on the storage of exemplars, and
- processing is based on the retrieval of exemplars, or for similarity-based reasoning, on the basis of exemplars.

IB	all chunk-initial words receive a B tag
IE	all chunk-final words receive a E tag
IBE1	all chunk-initial words receive a B tag, all chunk-final words receive a E tag; if there is only one word in the chunk, it receives a B tag
IBE2	all chunk-initial words receive a B tag, all chunk-final words receive a E tag; if there is only one word in the chunk, it receives a E tag

Table 1: Data Representation for Chunking

MBL can be used simply and effectively to perform a range of classification tasks.

3.4 Feature Selection

Feature selection is important for the performance for both machine learning techniques. In practice, the features we used are shown in Table 2. The information we used was contained in a 7-word window, i.e. the leftmost three words and their Part-of-Speech (POS) tags, the current word and its POS tag, and the rightmost three words and their POS tags.

3.5 Multi-level Chunking

3.5.1 Notation

Given a sentence s_1^I containing I words $\{w_1, \ldots, w_I\}$, chunking can be considered as the process of inserting a chunk boundary marker c_i between two consecutive words w_i, w_{i+1}. The probability of inserting a chunk boundary marker c_i between two consecutive words w_i, w_{i+1} (i.e. the partition probability) can be defined as:

$$\mathbb{p}(c_i|s_1^I) = p_{\lambda_1^M}(c_i|s_1^I)$$
$$= \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(c_i, s_1^I)]}{\sum_{c_i'} \exp[\sum_{m=1}^{M} \lambda_m h_m(c_i', s_1^I)]}$$

For sentence s_1^I, we can derive a set of partition probabilities with $I-1$ elements:

$$PP = \{\mathbb{p}(c_1|s_1^I), \ldots, \mathbb{p}(c_{I-1}|s_1^I)\}$$

By setting different thresholds for our partition probabilities, we can obtain different chunking results for the same sentence. This threshold can be adjusted depending on the task at hand with the result that different chunking patterns for the same sentence are obtained. We call this chunking model *Multi-level Chunking*.

If we relate this model to our IE data representation (cf. (2) above), it is equivalent to determining the probability of a word being labelled E. While most chunking approaches are essentially classification-based, our model attempts to transform the classification-based approach into a ranking problem and decide the partition point of a sentence by examining competitive scores at each point. We call this chunking approach *Ranking-based Chunking*.

The set of parameters in this model include (i) the set of partition probabilities, and (ii) estimates of thresholds for partition probabilities bearing in mind the specific task to be performed.

Figure 2 gives an example of the distribution of the partition probability.

The ||| city ||| bears ||| the ||| weight ||| of ||| powerful |||
0.7069 0.5307 0.5467 0.4527 0.3777 0.4098 0.4162
symbols ||| for ||| all ||| three ||| monotheistic ||| religions ||| .
0.4318 0.4253 0.3807 0.5655 0.5078 0.9796

Figure 2: Example of Multi-level chunking

If we take 2 words as our average chunk length, we can chunk sentence (2) as shown in Figure 3.

The ||| city ||| bears ||| the ||| weight of powerful symbols for all three ||| monotheistic religions ||| .

Figure 3: Example of chunking result using Multi-level chunking

Note that several words *weight ... three* have been combined into one chunk in Figure 3 based on the partition probabilities shown in Figure 2.

Word	w_{i-3}	w_{i-2}	w_{i-1}	w_i	w_{i+1}	w_{i+2}	w_{i+3}
POS	t_{i-3}	t_{i-2}	t_{i-1}	t_i	t_{i+1}	t_{i+2}	t_{i+3}

Table 2: Features for chunking

3.5.2 Threshold Estimation

The average length of chunks can be estimated from training data acquired following the criteria described in Section 3.1. With an estimation of average chunk length, we can set a chunking threshold to chunk a sentence.

4 Experimental Setting

4.1 Evaluation

Using the Alignment-Guided Chunking approach described in Section 3, we can train two different chunkers on French–English (FE-Chunker) and German–English (DE-Chunker) bilingual corpora respectively. We use the two chunkers to perform chunking on the same English sentences. Two test sets are constructed, each suitable for the FE-Chunker and the DE-Chunker respectively. The performance of the two chunkers is evaluated on the appropriate test set.

4.2 Gold Standard Test Set

For each sentence E in the test set, there could be N translation references r_1^N. For each sentence pair $< E, r_i >$, a unique word alignment A_i can be acquired. Following the criteria described in Section 3.1, we can derive N chunking results C_1^N using $< E, A_i >$ $(i \in [0, N])$. All these chunking results should be considered to be correct. Chunking results for E using our approach are evaluated on C_1^N using just one 'gold standard' reference.

We firstly construct the test set automatically using the criteria described in Section 3.1. After that we check all the sentences manually to correct all the chunking errors due to word alignment errors.

4.3 Data

The experiments were conducted on French–English and German–English sections of the Europarl corpus (Koehn, 2005) Release V1.[1] This corpus covers April 1996 to December 2001, and we use the Q4/2000 portion of the data (2000-10 to 2000-12) for testing, with the other parts used for training. The English sentences in the French–English and German–English corpora are not exactly the same due to differences in the sentence-alignment process. We obtain the intersection of the English sentences and their correspondences to construct a new French–English corpus and German–English corpus, where these two corpus now share exactly the same English sentences.

In order to test the scalability of our chunking approach, we first use 150k of the sentence pairs for training, which we call the *Small Data* set. Then we use all the sentence pairs (around 300k sentence pairs) for training. We call this the *Large Data* set.

We tag all the English sentences in the training and test sets using a maximum entropy-based Part-of-Speech tagger-MXPOST (Ratnaparkhi, 1996), which was trained on the Penn Treebank (Marcus et al., 1993). We use the GIZA++ implementation of IBM word alignment model 4 (Brown et al., 1993; Och and Ney, 2003)[2] and refinement heuristics described in (Koehn et al., 2003) to derive the final word alignment.

We used the Maximum Entropy toolkit 'maxent',[3] and the Memory-based learning toolkit TiMBL[4] for parameter estimation.

4.4 Statistics on Training Data

To demonstrate the feasibility of adapting our chunking approach to different languages, we obtained some statistics on the chunks of two training sets derived from French–English (F-E, 300k-sentence pairs) and German–English

[1] http://people.csail.mit.edu/koehn/publications/europarl/

[2] More specifically, we performed 5 iterations of Model 1, 5 iterations of HMM, 5 iterations of Model 3, and 5 iterations of Model 4.

[3] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

[4] http://ilk.uvt.nl/timbl/

(D-E, 300k-sentence pairs) corpora respectively. There are 3,316,887 chunks identified in the F-E corpus and 2,915,325 chunks in the D-E corpus. A number of these chunks overlap: 42.08% in the F-E corpus and 47.87% in the D-E corpus (cf. Table 3). The number of overlapping chunks (OL chunks) between these two corpora is 1,395,627.

	F-E	D-E
No. of Chunks	3,316,887	2,915,325
OL Chunks[%]	42.08%	47.87%

Table 3: chunk statistics

We can also estimate the average chunk length on training data. Using the F-E corpus, the average chunk length for English is 1.84 words and 2.10 words using the D-E corpus. This demonstrates definitively that our approach does carve up sentences differently depending on the target language in question.

5 Experimental Results

5.1 Results

Two machine learning techniques—Maximum Entropy (MaxEnt) and Memory-based learning (MBL)—are used for chunking. In order to test the scalability of our chunking model, we carried out experiments on both the Small data and Large data sets described in Section 4.3.

The detailed results are shown in Table 4. Here we can see that the F-score is quite low because we have just one reference in the test set (see Section 4.2). Furthermore, we see no significant improvement with the maximum entropy method when more data is used.

F-scores for German chunks are on the whole between 25 and 33% higher than for French. For German, when using MaxEnt Precision scores are significantly higher than Recall, but the opposite is seen when MBL chunks are used. For French, Recall scores are higher in general than those for Precision.

Figure 4 gives an example of chunking results using MaxEnt. Note the differences between this output and that in Figure 3: the determiner *the* has now been properly grouped with the following N-bar *weight of powerful symbols ...*, and similarly *all* belongs more closely to *three monotheistic religions* than it did before.

The ||| city ||| bears ||| the weight of powerful symbols for ||| all ||| three ||| monotheistic ||| religions ||| .

Figure 4: Example of chunking result

5.2 Multi-level Chunking

As an extension to our classification-based chunking method, multi-level chunking can be regarded as an application of ranking. We obtain the global chunk length from the training data to derive the optimal partition threshold. We use the average chunk length from the training data described in Section 4.4, i.e. for the French–English task, the average English chunk length is 1.84 words, whereas it is 2.10 words for German–English. The results of applying the multi-level chunking method (Multi) are shown in Table 5.

By using the multi-level chunker, we can see a slight increase in recall together with a sharp decrease in precision. This demonstrates that deriving chunks using just a global average chunk length is likely to be sub-optimal for any given sentence.

6 Conclusions and Future Work

In this paper, we have introduced a novel chunking approach guided by the word alignment acquired from bilingual corpora. We investigate our chunking approach on two language pairs: French–English and German–English, where these two bilingual corpora share the same English sentences. Two machine learning techniques—Maximum Entropy and Memory-based learning—were employed to perform chunking. We demonstrate the impact of chunking results on the English side due to the differences between French–English word alignment and German English word alignment, demonstrating the merit of such a chunking approach in a bilingual context. We evaluate the performance of our chunking approach on a one-reference gold standard test set and report an F-score

	Accuracy		Precision		Recall		F-score	
	FR	DE	FR	DE	FR	DE	FR	DE
MaxEnt-Large	55.37	68.41	30.89	47.57	34.57	35.12	32.63	40.41
MBL-Large	52.70	65.75	24.08	38.00	30.43	41.61	26.88	39.72
MaxEnt-Small	55.08	68.37	30.83	47.37	35.26	34.93	32.90	40.21
MBL-Small	52.53	65.56	23.96	37.62	30.41	40.83	26.80	39.16

Table 4: Results of Classification-based Chunking[%]

	French			German		
	Precision	Recall	F-score	Precision	Recall	F-score
MaxEnt	30.89	34.57	32.63	47.57	35.12	40.41
MBL	24.08	30.43	26.88	38.00	41.61	39.72
MaxEnt-Multi	28.41	34.69	31.24	38.14	38.11	38.12
MBL-Multi	22.69	28.18	25.14	34.36	38.46	36.29

Table 5: Classification-based Chunking vs. Ranking-based Chunking[%]

of 32.63% for the *FE-Chunker* and 40.41% for the *DE-Chunker*. We also extend our chunking approach with *Multi-level Chunking*, which is more tolerant of the chunking errors, but lower Precision scores are seen across the board.

As for future work, we want to experiment with other methods of word alignment (e.g. (Tiedemann, 2004; Liang et al., 2006; Ma et al., 2007)) in order to establish which one is most appropriate for our task. We also want to apply this method to other corpora and language pairs, especially using IWSLT data where for 4 language pairs we have 16 reference translations. We anticipate that our chunking approach is likely to be of particular benefit, at least in theory, in a statistical machine translation task given the complexities of the decoding process. Nonetheless, the principal remaining concern is whether the better motivated yet considerably smaller number of bilingual chunks derived via our method will lose out in a real task-oriented evaluation compared to a baseline system seeded with phrase pairs produced in the usual manner.

Acknowledgments

This work is supported by Science Foundation Ireland (grant number OS/IN/1732). We would also like to thank the anonymous reviewers whose insightful comments helped improve this paper.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* **19**(2):263–311.

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computatioanl Linguistics* **22**(1):39–71.

Walter Daelemans and Antal van den Bosch. 2005. *Memory-Based Language Processing.* Cambridge University Press, Cambridge, UK.

Nano Gough and Andy Way. 2004. Robust large-scale EBMT with marker-based segmentation. In *Proceedings of the 10th Conference on Theoretical and Methodological Issues in Machine Translation (TMI-04)*, Baltimore, MD., pp.95–104.

T. Green. 1979. The necessity of syntax markers: two experiments with artificial languages. *Journal of Verbal Learning and Behavior* **18**:481–496.

Declan Groves and Andy Way. 2005. Hybrid example-based SMT: the best of both worlds? In *Proceedings of the workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond, ACL 2005*, Ann Arbor, MI., pp.183–190.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X*, pp.79–86, Phuket, Thailand.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pp.48–54, Edmonton, Canada.

Rob Koeling. 2000. Chunking with Maximum Entropy Models. In *In Proceedings of CoNLL-2000*, Lisbon, Portugal, pp.139–141.

Percy Liang, Ben Taskar and Dan Klein. 2006. Alignment by Agreement. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006)*, New York City, NY., pp.104–111.

Feifan Liu, Qianli Jin, Jun Zhao and Bo Xu. 2004. Bilingual chunk alignment based on interactional matching and probabilistic latent semantic indexing. In *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04)*, Sanya, Hainan Island, China, pp.416–425.

Yanjun Ma. 2006. *Automatic Identification and Alignment of Chinese-English Phrases based on Multi-strategies*. MA thesis, Tsinghua University, Beijing, China.

Yanjun Ma, Nicolas Stroppa and Andy Way. 2007. Bootstrapping Word Alignment Via Word Packing. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07)*, Prague, Czech Republic, pp.304–311.

Michell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* **19**(2):313–330.

Franz Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics* **29**(1):19–51.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the third ACL Workshop on Very Large Corpora*, Somerset, NJ., pp.82–94.

Adwait Ratnaparkhi. 1996. A Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 1996)*, Philadelphia, PA., pp.133–142.

Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task: chunking. 2000. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, pp.127–132.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL 1999)*, Bergen, Norway, pp.173–179.

Nicolas Stroppa and Andy Way. 2006. Matrex: DCU Machine Translation System for IWSLT 2006. In *Proceedings of IWSLT 2006 Workshop*, Kyoto, Japan, pp.31–36.

Jörg Tiedemann. 2004. Word to Word Alignment Strategies. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, Geneva, Switzerland, pp.212–218.

Wei Wang, Ming Zhou, Jinxia Huang, and Changning Huang. 2002. Structure alignment using bilingual chunking. In *Proceedings of COLING 2002*, Taipei, Taiwan.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpus. *Computational Linguistics* **23**(3):377–403.

BLEUÂTRE: Flattening Syntactic Dependencies for MT Evaluation

Dennis N. Mehay
Department of Linguistics
The Ohio State University
Columbus, OH, USA
mehay@ling.osu.edu

Chris Brew
Department of Linguistics
The Ohio State University
Columbus, OH, USA
cbrew@ling.osu.edu

Abstract

This paper describes a novel approach to syntactically-informed evaluation of machine translation (MT). Using a statistical, treebank-trained parser, we extract word-word dependencies from reference translations and then compile these dependencies into a representation that allows candidate translations to be evaluated by string comparisons, as is done in n-gram approaches to MT evaluation. This approach gains the benefit of syntactic analysis of the reference translations, but avoids the need to parse potentially noisy candidate translations. Preliminary experiments using 15,242 judgments of reference-candidate pairs from translations of Chinese newswire text show that the correlation of our approach with human judgments is only slightly lower than other reported results. With the addition of multiple reference translations, however, performance improves markedly. These results are encouraging, especially given that our system is a prototype and makes no essential use of synonymy, paraphrasing or inflectional morphological information, all of which would be easy to add.

1 Introduction

Effective automatic translation evaluation (ATE) systems are crucial to the development of machine translation (MT) systems, as the relative performance gain of each minor system modification must be tested quickly and cheaply. A professional human evaluation of MT system output after each such modification is too expensive and time-consuming for rapid, cost-effective deployment of translation software.

For the past few years, n-gram precision metrics for MT evaluation such as BLEU (Papineni et al., 2002) and the related NIST metric (Doddington, 2002) have been the standard approach to ATE. In essence, BLEU and NIST measure the quality of a candidate translation as a function of the number of n-grams (typically, $1 \leq n \leq 4$) it shares with a set of (one

or more) reference translations. These metrics require a one-time investment of creating a reference corpus of translations to test the system against, but are fully automatic once this corpus has been created and are very portable, requiring only word tokenisers for the reference set (if it is not already tokenised).

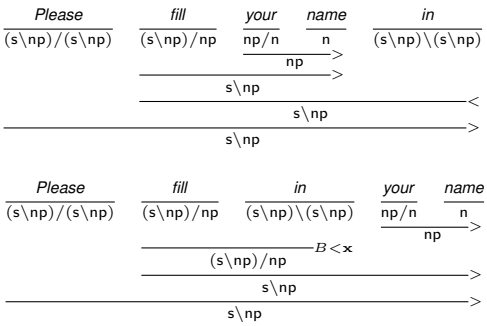
The portability of n-gram-based models, however, is one side of a trade-off with robustness: candidate translations are rewarded or penalised according to how well they match the *exact, contiguous word sequences in the reference set*. Candidates that contain legitimate word order variation will be penalised for not having these exact matches. Increasing the size of the reference set so as to capture more translational variation (as suggested by Thompson (1991)) is one possibility, but this is an expensive and time-consuming alternative. Moreover, given that adjuncts (e.g., adverbial modifiers), stacked attributive adjectives and a host of other grammatical elements can often "move around" without significantly affecting the meaning of a sentence, the strategy of padding the reference set with more examples for a word n-gram approach can only accommodate a fraction of the legitimate, syntactically-licensed variation in word order that a candidate translation should be allowed to display.

It seems reasonable, then, to explore approaches to ATE that exploit syntactic information so as not penalise legitimate syntactic variation. This paper describes such an approach. We describe here a prototype system called BLEUÂTRE[1] ("bluish"), a novel approach to syntactically-informed automatic machine translation evaluation that uses syntactic word-word dependencies from parses of reference translations. In this approach, we use a statistical Combinatory Categorial Grammar parser (Clark and Curran, 2004) to parse the reference set and extract word-word dependencies based on hierarchical head-dependent relationships (or "tectogrammatical" relationships). These dependencies are then compiled out into bags of dependent words that must appear to the left and right of each head word — essentially enforcing a partial linear ordering of dependents with respect to their heads. The quality of a candidate translation is then evaluated according to the number of these head word-dependent word partial orderings that it recalls. This approach is novel in that it only requires parses of reference translations, avoiding the need to parse (potentially noisy) candidate translations.

Preliminary experiments using 15,242 judgments of reference-candidate pairs from translations of Chinese newswire text show that BLEUÂTRE's correlation with human judgments is competitive with, but lower than, other reported results. With the addition of multiple reference translations for each system judgment, however, performance improves markedly. These results are encouraging, especially given that BLEUÂTRE is a prototype and makes no essential use of synonymy, paraphrasing or inflectional morphological information. The essential contribution of this paper is a description of how syntactic dependencies can be "flattened" to a form suitable for evaluating unparsed candidate translation sentences. We anticipate that this approach can be profitably combined with other syntactic and non-syntactic approaches to ATE.

The remainder of this paper is organised as follows: Section 2 describes how we use the parser to extract dependencies and how BLEUÂTRE uses these dependencies for eval-

[1] Standing for **BLEU**'s **A**ssociate with **T**ectogrammatical **RE**lations.



(det name$_3$ your$_2$)	(det name$_4$ your$_3$)
(dobj fill$_1$ name$_3$)	(dobj fill$_1$ name$_4$)
(ncmod _ fill$_1$ in$_4$)	(ncmod _ fill$_1$ in$_2$)
(xcomp _ please$_0$ fill$_1$)	(xcomp _ please$_0$ fill$_1$)

Figure 1: A CCG derivations and corresponding dependency graphs for the word order variants *Please fill your name in* and *Please fill in your name*.
(Key: det=‘determiner’, dobj=‘direct object’, ncmod=‘non-clausal modifier’ and xcomp=‘externally controlled clausal complement’.)

uation. Section 3 describes related work. Section 4 describes our preliminary experiments, and Section 5 is a conclusion that also briefly outlines future work.

2 Extracting and Using Dependencies for ATE

In our experiments, we use a statistical Combinatory Categorial Grammar (CCG) parser (Clark and Curran, 2004). CCG (Steedman, 2000) is a “mildly context-sensitive” formalism that provides elegant analyses of coordination (including “non-constituent” coordination), extraction, right node raising and other constructions that have proved challenging in other frameworks.

Figure 1 illustrates the CCG derivation and corresponding Briscoe and Carroll-style grammatical role dependencies that the Clark and Curran (C&C) parser outputs for the sentence *Please fill your name in*.[2] A parse of the semantically identical *Please fill in your name* would give the identical dependency graph (*modulo*, of course, the different string indices on the words).

Note, however that, if the first sentence is a reference translation and the second sentence is a candidate translation, then an n-gram-based approach to ATE would heavily penalise this minor variation in word order, even though it is identical both in syntactic dependency structure and semantic content. This is because, although the two sentences share all the same unigrams, the second sentence only contains two of the four bigrams from the reference sentence (and none of the 3-grams or 4-grams), giving it a relatively low BLEU score. A method that compared the overlap of the syntactic dependencies of the two sentences, however, would not penalise this minor word-order variation at all.

Note, however, that only a *correct* parse of the second sentence would give the identical dependency graph as the first. In fact the C&C parser, despite its state of the art performance,[3] does not parse this well-formed sentence correctly. Instead, due to part-of-speech tagging errors, it improperly treats ‘*in*’ as a preposition and not a particle, giving a parse that treats ‘*in your name*’ as a PP modifying a non-phrasal verb ‘*fill*’. This induces the following (incor-

[2] For the uninitiated, the horizontal (underlining) lines are analogous to branchings in a traditional tree representation of a syntactic derivation, where the $\ldots < \ldots$ and $\ldots > \ldots$ annotate the direction and type of the combinatory mechanism that produced each such “branching”.

[3] With $\approx 85\%$ balanced F-score in recovering both local and long-distance labelled dependencies.

rect) dependency graph:

(det name$_4$ your$_3$)
(dobj fill$_1$ in$_2$)
(dobj _ in$_2$ name$_4$)
(xcomp _ please$_0$ fill$_1$)

Ignoring the errors in the labels of the dependency arcs, we can see that the *unlabelled* dependency structure is also wrong: the direct dependency between '*fill*' and '*name*' is lost.

The fact that parsers can and often do err on well-formed sentences suggests that their performance will degrade considerably on less well-formed MT system output. This motivates the principle innovation of BLEUÂTRE: namely, we compile out the dependency triples from the parse of a candidate translation into bags of dependent words that must appear either to the left or right of each head word. This is essentially a partial linear ordering of dependents with respect to their heads. The essential point of this approach is that it avoids parsing MT system output. The following illustrates this process on our hypothetical reference sentence *Please fill your name in*:

$\emptyset$	$\overleftarrow{left}$	'**Please**'	$\overrightarrow{right}$	{'fill'}
$\emptyset$	$\overleftarrow{left}$	'**fill**'	$\overrightarrow{right}$	{'in','name'}
{'your'}	$\overleftarrow{left}$	'**name**'	$\overrightarrow{right}$	$\emptyset$

These partial orderings of dependents — which we shall sometimes call "*left and right contexts*" — allow candidates to be evaluated by a simple string search, verifying whether each of the dependents is either to the right or to the left of the head word as the case may be. The score of a candidate with respect to a reference is the number of such left-right orderings that it recalls multiplied by an exponentially decaying "length penalty", which is inspired by BLEU's brevity penalty. The intuition is that, the longer a candidate translation is, the more of the reference dependency orderings it is likely to recover, and, thus, candidate sentences longer than the reference must be penalised. Candidates shorter than the reference, in effect, penalise themselves, as they do not contain as many words that could match those in the left-right contexts, and, as such, no brevity penalty is assessed. In symbols, a candidate c's dependent ordering score for a single head word h that is in the reference r is the following:

$$\text{DEP}_{c,h,r} = \sum_{d_i \in \text{lf}(h)} \Lambda_c(d_i, h) + \sum_{d_j \in \text{rt}(h)} \rho_c(d_j, h)$$

where c is the candidate translation, $\text{lf}(h)$ is the left context of h in r, $\text{rt}(h)$ is the right context of h in r, and the functions $\Lambda_c(d_i, h)$ and $\rho_c(d_j, h)$ have value 1 if both $h \in c$ and d_i (or d_j, respectively) is to the left (or right) of h in c, and 0 otherwise.[4]

The BLEUÂTRE recall score of a candidate c with respect to a reference r is then:

$$\text{BLEUÂTRE}_{c,r} = LP_{c,r} \cdot \Big(\frac{\sum_{h \in r} \text{DEP}_{c,h,r}}{\sum_{h \in r} |\{d : d \in lf(h) \vee d \in rt(h)\}|} \Big)$$

Where $LP_{c,r}$, the length penalty of a candidate with respect to a reference, is simply BLEU's brevity penalty with the roles of the candidate and reference lengths reversed:

$$LP_{c,r} = \begin{cases} 1, \text{ if } len(c) < len(r) \\ e^{\left(1 - \frac{len(c)}{len(r)}\right)}, \text{ otherwise} \end{cases}$$

As a concrete example, take our hypothetical candidate translation *Please fill in your name*. This candidate scores a perfect 1.0, because

[4] Essentially, these functions signal whether the dependent is properly ordered with respect to the head in the candidate translation.

'*fill*' is to the right of '*Please*', '*in*' and '*name*' are to the right of '*fill*' and '*your*' is to the left of '*name*', and the sentences have the same length. Thus the syntactically licit word order variation is not penalised. Imagine further a less well-formed candidate translation from Dutch '*Vul even uw naam in*' $\Rightarrow$ '*Fill please your name in*'. Even though this candidate has only 1 bigram (and no 3- and 4-grams) in common with the reference (thus, giving it a low BLEU score), it still receives a fairly high BLEUÂTRE score of 0.75, since only '*please*' and '*fill*' are out of the order specified by the parse of the reference. This accords with our intuitions that '*Fill please your name in*' is only mildly "Dutch-sounding" and conveys the gist of the reference.

3 Related Work

There is a growing concern in the MT research community as to the correlation of BLEU with human judgments of translation quality, even at the document level (Callison-Burch et al., 2006). This is of particular concern, as statistical MT systems are now trained to minimise error with respect to ATE metrics (Och, 2003).

There have been many attempts to improve upon the performance of BLEU. The NIST metric mentioned above (Doddington, 2002) uses n-gram precision scores as BLEU does, but it weights the information contributed by certain n-grams. In this approach, rare n-grams count more than frequent n-grams in a candidate's precision score. Turian et al.'s (2003) approach (called General Text Matcher or GTM) is to compute both precision and recall of a candidate's match to the reference set, scoring contiguous sequences higher than discontiguous matches. Kulesza and Shieber (2004) describe a machine learning-based approach to combining various metrics such as BLEU-style n-gram precision ($1 \leq n \leq 5$), word error rate, position-independent word error rate, etc. These values are passed as features to a support vector machine (Vapnik, 1995) which learns to discriminate human from machine-generated translations. The farther a candidate translation's feature encoding is on the human side of the hyperplane separating human from machine translations, the better it is judged to be.

(Banerjee and Lavie, 2005) describes METEOR, a word-based generalised unigram matching approach that rewards sentence alignments between references and candidates that minimise the number of crossing word alignments. Stemming and WordNet synonyms are used to improve the match between translations that may differ only in their lexical choice or grammatical use of a particular base word form. All of these approaches, however, are still based on matching a candidate to a reference at the word level, and, as such, they are ultimately still susceptible to reduced performance due to syntactically acceptable variation.

Thus, some authors have attempted to use syntactic information in ATE. Liu and Gildea (2005) parse both reference and candidate translations. The count of subtrees up to a fixed, uniform depth that the candidate recalls is one metric used. Also, by decomposing each parse tree into a vector of counts of all subtrees, the authors compute the cosine between the reference and candidate vectors. Both metrics are also computed for dependency parses, as extracted from the phrase-structure parses of the candidate and reference translations. Finally, the authors compute the fraction of dependency chains (up to some fixed length) in the reference that are also in the candidate. The authors report improved correlation with human judgments as compared with BLEU.

Recently, Owczarzak et al. (2007) have reported using Lexical Functional Grammar (LFG) grammatical functional dependency triples to evaluate translation quality. Their approach is also to parse both the reference and candidate translations. They directly compute the dependency precision and recall of the candidate translation with respect to the reference. These authors perform an extensive comparison of their system to various ATE metrics over the Linguistic Data Consortium's Multiple Translation Chinese corpus (parts 2 and 4). When supplementing the dependency matches with WordNet synonyms, they achieve the highest correlation to human judgments in fluency and second place in an average of fluency and accuracy, as compared to BLEU, NIST, GTM, Translation Error Rate (TER, (Snover et al., 2006)) and METEOR. We have used this same corpus and, as such, can compare our results to theirs, as well as the other approaches they tested over this corpus. Our approach is distinguished from these last two approaches in that we do not attempt to parse candidate translations.

4 Preliminary Experiments

To test our system, we used sections 2 and 4 of the TIDES 2003 Chinese-to-English Multiple Translation corpus (MTC) of newswire text (released by the LDC). This corpus contains various commercial off-the-shelf (COTS) and research MT systems' translations of a set of Chinese source sentences. There are 4 human-produced reference translations for each source sentence. There are also human translation quality (fluency and accuracy) judgments for a subset of the machine-produced translations. We use these quality judgments to track the performance of BLEUÂTRE.

4.1 Experiment 1

The human judges were only shown a single "best" reference translation (as determined by an independent expert), and, so, following Owczarzak et al. (2007), we compute Pearson's correlation coefficient of the BLEUÂTRE score to each reference-candidate-judgment triple for our first experiment. This gives 15,242 total points of comparison (triples). This number is less than the 16,800 triples used by Owczarzak et al. (2007), as the C&C parser was only able to find a spanning analysis for 98.2% of the reference sentences, and many of these reference sentences are used several times as a gold standard for the human evaluators.[5]

The results of BLEUÂTRE's correlation to human fluency, accuracy and an average of the two are displayed in Table 1. To the extent that our approach is comparable with the results in (Owczarzak et al., 2007), we have listed their relevant results for comparison. Note that TER is negatively correlated with human judgments. This is because 0 is a perfect TER score. Owczarzak et al.(2007) note, however, that this still allows comparison of the absolute values of the correlation coefficients. Our system uses word-word dependencies, with no recourse to external morphological or thesaurus-based resources, such as WordNet. We therefore compare only with systems that use the same type of input. Future work may use a wider range of lexical resources and allow a wider range of meaningful comparisons.

We note that BLEUÂTRE does as well as

[5]The parser employs a back-off strategy that expands the parse search space incrementally to five back-off levels. After five unsuccessful back-off retries, however, the parser returns a failure notice and moves on to the next sentence. These settings are the off-the-shelf settings of the C&C parser with an additional, less-restrictive back-off level, as well as with a larger maximum size on the parse chart.

FL	HAC	AVE
BLEU 0.155*	MET 0.278*	MET 0.242*
OEtAl 0.154*	NIST 0.273*	NIST 0.238*
MET 0.149*	GTM 0.260*	OEtAl 0.236*
NIST 0.146*	OEtAl 0.224*	GTM 0.230*
GTM 0.146*	BA 0.202	BLEU 0.197*
TER -0.133*	BLEU 0.199*	BA 0.186
BA 0.128	TER -0.192*	TER -0.182*

Table 1: Pearson's correlation between various evaluation metrics and human judgments. BLEUÂTRE's results are our own. * indicates that the results are as reported in (Owczarzak et al., 2007) for the same set of reference-candidate-judgment triples (modulo C&C parsing failures). (Key: **BA**=BLEUÂTRE; **OEtAl**=Owczarzak et al.'s "predicate-argument dependency" system; **MET**=METEOR without WordNet or stemming; **FL**= Human fluency judgments; **HAC**=human accuracy judgments; **AVE**=Average of FL and HAC. Other abbreviations are given above.)

TER in fluency and both TER and BLEU in accuracy and fluency-accuracy average.[6]

Perhaps surprisingly, BLEUÂTRE correlates better with human accuracy judgments than with fluency judgments. We would expect approaches that pay appropriate attention to syntax to do well on fluency, because it is closely associated with grammatical well-formedness. We suspect that that BLEUÂTRE is still too conservative about word order variation. It seems to over-enforce partial orderings of dependents with respect to their heads [7]. It appears that human raters are better able to overlook this kind of variation, and that this emerges in their fluency judgments.

[6]Only a change of 0.015 or greater is significant at the 95% confidence level for both ours and Owczarzak et al.'s (2007) results.

[7]E.g. "*Fill your name in, please*" does not satisfy the partial (right-hand side) ordering of '*fill*' to '*Please*' as extracted from our hypothetical reference translation above.

FL	HAC	AVE
UFS 0.143	BA 0.208	BA 0.190
LFS 0.142	UFS 0.196	UFS 0.189
BA 0.130	LFS 0.194	LFS 0.188

Table 2: Pearson's correlation between BLEUÂTRE, and C&C parser-based f-score evaluation (labelled and unlabelled). Key: **BA**=BLEUÂTRE; **LFS**=Labelled F-score; **UFS**=Unlabelled F-score; (correlations to) **FL**=Human fluency judgments; **HAC**=human accuracy judgments; **AVE**=Average of FL and HAC. Only a difference of ± 0.016 is significant with 95% confidence (no significant differences).

4.2 Experiment 2

An obvious question raised by the above results is whether our decision not to parse candidate translations is helpful — it may be that the differences between Owczarzak et al. (2007)'s results and ours are not due to this feature of the system but rather to other differences such as the nature of the parsers or grammatical formalisms used (LFG vs. CCG). To investigate this, we compare BLEUÂTRE's correlation to human judgments to that of a re-implementation of the Owczarzak et al. (2007) approach by computing the f-score between parses of the candidate translations and the corresponding reference translations using the C&C parser. We compute this score for both labelled and unlabelled dependencies and compare it with BLEUÂTRE's correlation to a subset of the reference-candidate-triples where

both BLEUÂTRE and the f-score methods were able to provide a score.[8] This results in a set of 14,138 scores by BLEUÂTRE and the f-score methods compared against reference-candidate-judgment triples.

Table 2 gives the correlation of BLEUÂTRE and the two f-score methods to the relevant 14,138 human judgments. Although BLEUÂTRE differs slightly from the other methods, none of the differences is statistically significant. This confirms our intuition that BLEUÂTRE is proving effective at extracting and applying syntactic criteria when assigning scores to candidate translations. In effect, it is an alternative means of doing the job for which (Owczarzak et al., 2007) use the parser.

4.3 Experiment 3

In a third experiment, we include multiple reference translations to provide more partial orderings, thus minimising BLEUÂTRE's sensitivity to partial orderings extracted from a single reference translation. For this, we simply compute BLEUÂTRE scores for each candidate-reference pairing and pick the highest score as the BLEUÂTRE multiple-reference score. Owczarzak et al. (2007) do not describe such an experiment, and so our results are not comparable to theirs. Liu and Gildea (2005), however, do perform such an experiment, as do Banerjee and Lavie (2005). Accordingly, we performed two sub-experiments for comparison with these authors' work:[9]

[8] As the C&C parser only achieves 98% coverage on the reference set and 91% on the test set, we compare BLEUÂTRE and the f-score approach on the intersection of the parsed reference and candidate examples.

[9] Keeping in mind that the data sets are not identical due to C&C parsing failures. These failures, however, only lead to a few instances where there is no parsable reference sentence for a candidate. 915 sentences in E14 and 910 sentences in E15 were given BLEUÂTRE scores. Liu and Gildea report having 925 sentences per section, which means we have a loss of coverage of 1% and 2%, respectively, on these sections.

E14-FL	**E15-FL**
BA 0.199	**BA** 0.188
LG_dt 0.159*	**LG_pt** 0.144*
LG_dc 0.157*	**LG_dt** 0.137*
LG_pt 0.147*	**LG_dc** 0.128*
BLEU 0.132*	**BLEU** 0.122*
LG_dtvc 0.090*	**LG_ptvc** 0.089*
LG_ptvc 0.065*	**LG_dtvc** 0.066*

Table 3: Correlation of BLEUÂTRE and Liu and Gildea's metrics to human fluency judgments for systems E14 and E15. (Key: * indicates that the score is from (Liu and Gildea, 2005); **BA**=BLEUÂTRE; **LG**=Liu and Gildea — different approaches: **_dt**=dependency subtrees, **vc**=vector-cosines, **_pt** structural subtrees; **_dc**=dependency chains.)

First, following Liu and Gildea (2005), we ran BLEUÂTRE to compute scores for systems E14 and E15 on part 4 of the Chinese Multiple Translation corpus using three reference translations (namely, those from E01, E03 and E04). We compare the segment-level BLEUÂTRE scores to human fluency scores for those same sentences.[10] We list these scores next to their best reported per-system scores (including their figures for BLEU over the same set) in Table 3.[11]

Second, we compute BLEUÂTRE scores individually for systems E09, E11, E12, E14, E15 and E22 (MTC, Part 4) using all four reference translations in E01-E04. We list the average

[10] Liu and Gildea also compute "overall" scores, which they describe as the sum of the fluency and accuracy score. We do not compare with these numbers.

[11] In our correlation tests, a difference of 0.06 is significant at the 95% confidence level. It is difficult to say how this compares with Liu and Gildea's results, but their data set is essentially the same as ours.

	BLEUÂTRE	METEOR
E09	0.338	0.351
E11	0.193	0.253
E12	0.216	0.264
E14	0.257	0.285
E15	0.238	0.237
E22	0.273	0.284
AVE	0.253	0.279

Table 4: BLEUÂTRE and METEOR's correlation to an average of human judgments of fluency and accuracy for various MT systems.

FL	HAC	AVE
0.235	0.328	0.315

Table 5: BLEUÂTRE correlation to across-judge human judgments using multiple references (MTC 2 and 4). Key: **FL**= Human fluency judgments; **HAC**=human accuracy judgments; **AVE**=Average of FL and HAC.

of these scores next to the relevant METEOR score (without WordNet or Porter stemming) in Table 4. This set of systems is different from those reported in (Banerjee and Lavie, 2005) — which also includes system E17 — as we do not have E17 in our LDC corpus. The METEOR scores were obtained by running METEOR (v 0.5) on the above-mentioned data.

These scores demonstrate that, with multiple reference translations, BLEUÂTRE's performance improves markedly and becomes competitive with other systems that report results using multiple references. It is notable that only a difference of ± 0.016 is significant with 95% confidence ($p \leq$ 3.609e-11) for both systems (BLEUÂTRE and METEOR). Thus, the difference in performance between our system and METEOR is not shown to be significant here.

Finally, for all judgments in MTC Parts 2 and 4, Table 5 gives BLEUÂTRE's correlation with an average of each of the human fluency and accuracy judgments, as well as to the average of the averages of each fluency-accuracy pair while using all four references. We are not aware of any study that has reported these figures. We simply offer them for comparison.

5 Conclusion and Future Work

We have shown that it is possible to extract syntactic dependency information from a reference translation and compile it to a form that allows candidate translations to be evaluated by simple string searches. While our approach currently does not achieve state-of-the-art performance with only one reference translation, we are encouraged by the fact that it is at least competitive with other methods such as TER and BLEU, and its performance is not significantly different from a direct parse-to-parse f-measure comparison on the same data set, using the same parser. Further, when BLEUÂTRE is allowed to maximise its score over multiple reference translations, its performance improves markedly. Here it is competitive with state-of-the-art approaches such as METEOR (v 0.5), and perhaps superior to more complicated syntax-based methods such as that in (Liu and Gildea, 2005), all while avoiding the overhead of parsing at evaluation-time.

A strength of our approach is that it is compatible with any parsing approach that outputs dependency triples and relative string positions. To improve the performance of our system, we would like to experiment with different parsers, as well as with stemming, electronic thesauri such as WordNet, and sources of synonymy and paraphrasing such as that described in (Owczarzak et al., 2006).

Finally, some dependencies (e.g. determiner-

noun dependencies) are unsurprising and perhaps "easier" to get right, so they should arguably not contribute much to assessments of progress in the field. We would like to explore schemes for using NIST-like weights to reward candidate translations for recalling more "valuable" dependencies such as, e.g., verb-object dependencies that are systematically missed by well-known benchmark systems.

6 Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful suggestions. Thanks also to Detmar Meurers for helpful feedback when BLEUÂTRE was a half-formed idea. The CCG parses in this paper were produced using Ben Wing's "wccg" extension to OpenCCG.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings the ACL*, Ann Arbor, MI, USA.

Chris M Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *Proceedings of the EACL-2006*, Trento, Italy.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the ACL*, Barcelona, Spain.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Human Language Technology Conference*, San Diego, CA, USA.

Alex Kulesza and Stuart M. Shieber. 2004. A learning approach to improving sentence-level MT evaluation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, Baltimore, MD, USA.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, MI, USA.

Franz Joseph Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the the ACL*, Sapporo, Japan.

Karolina Owczarzak, Declan Groves, Josef van Genabith, and Andy Way. 2006. Contextual bitext-derived paraphrases in automatic MT evaluation. In *Proceedings of the HLT-NAACL 2006 Workshop on Statistical Machine Translation*, New York, NY, USA.

Karolina Owczarzak, Josef van Genabith, and Andy Way. 2007. Dependency-based automatic evaluation for machine translation. In *Proceedings of the HLT-NAACL Workshop on Syntax and Structure in Statistical Translation*, Rochester, NY, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*, Philadelphia, PA, USA.

Matthew Snover, Bonnie Dorr, Richard Schwartz, and John Makhoul. 2006. A study of translation error rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas*, Cambridge, MA, USA.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, Massachusetts.

Henry Thompson. 1991. Automatic evaluation of translation quality: Outline of methodology and report on pilot experiment. In *(ISSCO) Proceedings of the Evaluators Forum*, Geneva, Switzerland.

Joseph P. Turian, Luke Shen, and I. Dan Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proceedings of MT Summit IX*, New Orleans, LA, USA.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin.

Demonstration of the Spanish to English METIS-II MT system

Maite Melero and Toni Badia
Grup de Lingüística Computacional (Barcelona Media), Barcelona, Spain
{toni.badia, maite.melero}@upf.edu

We present an experimental Machine Translation prototype system that is able to translate between Spanish and English, using very basic linguistic resources. In our approach, no structural transfer rules are used to deal with structural divergences between the two languages: the target corpus is the basis both for lexical selection and for structure construction. Our strategy emphasises modularity and language independence and, thus, is translatable to languages with very little NLP development.

Our system is currently being developed in the framework of Metis-II (Vandeghinste et al., 2006). The goal of the Metis project is to achieve corpus-based translation on the basis of a monolingual target corpus and a bilingual dictionary only. The bilingual dictionary functions as a flat translation model that provides n translations for each source word. The most probable translation given the context is then selected by consulting the statistical models built off the TL corpus[1].

Clearly, syntactic divergences between the source and target languages are among the major challenges that this minimalist translation strategy faces. Transfer systems typically address structural translation divergences via explicit bilingual mapping rules, either handwritten or example-based. In the Spanish-English prototype, we are able to do without a rule-based structural transfer component by handling translation divergences in the TL generation component.

By pushing the treatment of translation mismatches to the TL end component of the system, we make the treatment independent of the source language and consequently much more general. This solution is in line with other Generation intensive systems such as (Habash & Dorr, 2002). Like us, they are able to dispense with expensive sophisticated resources for the Source Language, however, unlike us, they need rich Target Language resources, such as lexical semantics, categorial variation and subcategorisation frames.

Our approach is also close to the work presented by (Carbonell et al., 2006). In their case, the output of the bilingual dictionary is decoded via long overlapping n-grams, built over full-form words; while we use non-overlapping n-grams over lemma-tag pairs. Also, in their system, in order to account for translation divergences, words and phrases in the SL and TL are substituted by synonyms and near-synonyms, which have been previously learned from TL and SL monolingual corpora.

For the preprocessing of the Spanish input, only very basic linguistic resources are needed, namely only a POS tagger and lemmatiser[2], whose output is a string of Spanish lemmas or base forms, with disambiguated POS tags and inflectional information. Morphological disambiguation is performed by selecting the most plausible reading for each word given the context. At a subsequent step, morphological tags are mapped into the Parole/EAGLES tagset[3] used by the bilingual dictionary. In this mapping step, information about POS, which will be used during dictionary look-up, is separated from inflectional information which will be used only later, in token generation.

Lexical translation is performed by a lemma-to-lemma dictionary, which contains information about the POS of both the source and the target word. The bilingual dictionary has been automatically extracted from a commercial machine readable dictionary, the Spanish-English Concise Oxford Dictionary (Rollin, 1998).

The output of the SL preprocessing and dictionary look-up is a set of translation candidates in form of strings of English lemmas and POS tags, ordered according to Spanish-like syntax.

As mentioned, translations that imply changes of structure are among the main difficulties of using a bilingual lexicon instead of a true translation model. These structure changes can ultimately be reduced to:

- local movement of Content Words (CW),
- deletion and insertion of Function Words (FW)[4], and

[1] The English corpus is a lemmatized version of the British National Corpus tagged using the CLAWS5 tagset. It contains over 6 million sentences.

[2] Our current tagger-lemmatiser is CastCG (Alsina et al., 2002), a shallow morphosyntactic parser for Spanish, based on the Constraint Grammar formalism.

[3] http://www.lsi.upc.es/$\sim$nlp/freeling/parole-es.html

[4] The following parts-of-speech are typically considered to be function words: articles, conjunctions, determiners, pronouns,

- movement of sentence constituents.

Our strategy, which makes crucial use of the distinction between function and content words, provided by the POS tagger, is based on the use of the target-language model to validate any change of structure occurring between SL and TL, instead of writing source-language dependent mapping rules.

A series of target language models are built by indexing all the n-grams for $1 \leq n \leq 5$[5]. An n-gram can belong to one of the following types:

- a sequence of lemma/tag (e.g. always/ADV + wear/VV + a/AT + hat/NN)
- a sequence of lemma/tag except for one position of tag alone(e.g. ADV + wear/VV + a/AT + hat/NN)

During the indexing process, tokens are usually indexed as either lemma/tag or tag alone. Exceptions are:

- personal pronouns (PNP) which are always lemma/tag
- cardinals (CRD), ordinals (ORD) and unknown words (UNC) which are always indexed as tag alone.

To account for structure modifications, we allow permutation of CWs between two consecutive boundaries[6], as well as insertion and deletion of a predefined set of FWs.

In the experiment described in (Melero et al. 2007), we compared the effect of each structure modifying operation in isolation and combined (see results in Table 1). It was run on a test corpus of 227 sentences, for which a set of 3 translation references per sentence was manually created by three independent translators.

Test set	Base	Ins	Del	Perm	All
Grammar	0.4698	0.4518	0.4746	0.4818	0.4658
News	0.3473	0.3358	0.3475	0.3687	0.3516
Technic	0.3072	0.2928	0.3085	0.3205	0.3038
Wiki	0.2720	0.2585	0.2720	0.2960	0.2789

Table 1: BLEU scores for the different settings

In this experiment, we chose as baseline the results of the search on the TL corpus with no structure changing operations. This baseline turned out to be quite high, probably because the word orders of the two languages involved are not extremely different. The variations of the different settings on this baseline are consequently small. The experiment shows the potential of the approach although also brings to light aspects that need to be addressed, such as optimization of weighs and scoring.

prepositions and, specific to English, the existential (*there*) and the infinitive marker (*to*).

[5] The 5-gram model is used only to build the Insertion and Deletion models.

[6] Boundary detection is performed on the basis of the POS information at hand. A boundary is defined by a pair of adjacent POS tags (e.g. NounArticle), which are considered to unambiguously indicate a transition between two consecutive constituents.

References

Alsina, A., Badia, T., Boleda, G., Bott, S., Gil, A., Quixal, M. and Valentí, O. (2002) CATCG: a general purpose parsing tool applied. In *Proceedings of Third International Conference on Language Resources and Evaluation*. Vol. III, pages 1130–1134, Las Palmas, Spain.

Carbonell, J., Klein, S., Miller, D., Steinbaum, M., Grassiany, T. and Frei, J. (2006) Context-based machine translation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Visions for the Future of Machine Translation*, pages 19–28, Cambridge, Massachusetts, USA.

Habash, N. and Dorr, B. (2002) Handling translation divergences: Combining statistical and symbolic techniques in generation-heavy machine translation. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, London, UK. Springer-Verlag.

Melero, Maite, Oliver, Antoni, Badia, Toni and Suñol, Teresa (2007) Dealing with Bilingual Divergences in MT using Target language N-gram Models. In *Proceedings of the METIS-II Workshop: New Approaches to Machine Translation. CLIN 17 - Computational Linguistics in the Netherlands*. (pp. 19-26) Leuven, Belgium

Rollin, N. (1998) The Concise Oxford Spanish Dictionary. Oxford University Press.

Vandeghinste, V., Schuurman, I., Carl, M., Markantonatou, S. and Badia, T. (2006) METIS-II: machine-translation for low-resource languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 1284–1289, Genoa, Italy.

Combining Resources for Open Source Machine Translation

Eric Nichols,[♯] **Francis Bond,**[♮] **Darren Scott Appling,**[♭] **Yuji Matsumoto**[♯]
[♯] Graduate School of Information Science, Nara Institute of Science and Technology
{eric-n,matsu}@is.naist.jp
[♮]National Institute of Information and Communications Technology
bond@ieee.org
[♭] College of Computing, Georgia Institute of Technology
darren.scott.appling@gatech.edu

Abstract

In this paper, we present a Japanese→English machine translation system that combines rule-based and statistical translation. Our system is unique in that all of its components are freely available as open source software. We describe the development of the rule-based translation engine including transfer rule acquisition from an open bilingual dictionary. We also show how translations from both translation engines are combined through a simple ranking mechanism and compare their outputs.

1 Introduction

While there have been many advances in the field of machine translation, it is widely acknowledged that current systems do not yet produce satisfactory results. At the same time, many researchers also recognize that no single paradigm solves all of the problems necessary to achieve high coverage while maintaining fluency and accuracy in translation (Way, 1999). It is our position that translation is a problem of meaning preservation, and that deep NLP is essential in meeting goals of high quality translation.

Our ultimate aim is to have a robust, high quality and easily extensible Japanese↔English machine translation system. Current stochastic MT systems are both robust and of high quality, but only for those domains and language pairs where there is a large amount of existing parallel text. Changing the type of the text to be translated causes the quality to drop off dramatically (Paul, 2006). Quality is proportional to the log of the amount of training data (Och, 2005), which makes it hard to quickly extend a system. Rule-based systems can also produce high quality in a limited domain (Oepen et al., 2004). Further, it is relatively easy to tweak rule-based systems by the use of user dictionaries (Sukehiro et al., 2001), although these changes are limited in scope.

Our approach to producing a robust, high quality system is to concentrate on translation quality and system extensibility, without worrying so much about coverage. We are able to do this because of the availability of a robust open source statistical machine translation systems (Koehn et al., 2007). As long as we can produce a system that produces good translations for those sentences it can translate, we can fall back on the SMT system for sentences that it cannot translate.

This leaves the problem of how to build a system that is both high quality and easily extensible. To gain high quality, we accept the brittleness of a rule-based semantic transfer system. In particular, by using a precise grammar in generation we ensure that the output is (almost always) grammatical. Rule types are hand-made. As far as possible we share types with the Norwegian→English system developed in the LOGON project (Oepen et al., 2004). To make the system (relatively) easily extensible, we construct transfer rules instances from a plain bilingual dictionary. As far as possible, we aim to concentrate our rule building efforts on closed-class words, and then fill in the open class transfer rules by automatic conversion

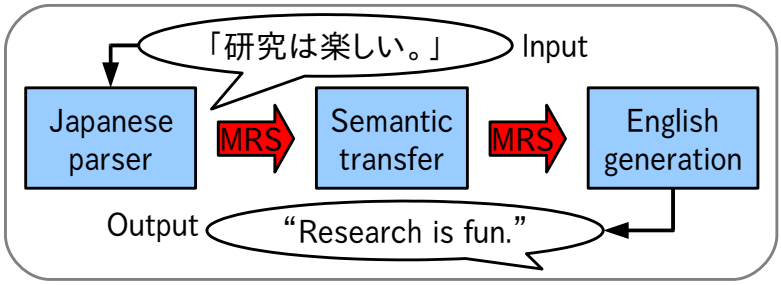


Figure 1: The Jaen machine translation architecture

of the bilingual lexicon. Finally, in future work, we will learn extra rules from aligned corpora.

In order to make this possible, we are working with an existing large scale collaborative Japanese-multilingual dictionary project (JMdict: Breen, 2004).

This paper is organized as follows. In Section 2, we present related research. In Section 3, we outline the development of our core system, and we introduce the DELPH-IN machine translation initiative that provided the resources used in its construction. In Section 4 we describe the expansion of our prototype system to target the Japanese-English section of the ATR Basic Travel Expression Corpus (BTEC*). In Section 5 we outline its integration with the Moses statistical machine translation system, and we compare translation results of these two systems in Section 6. We briefly discuss future work in Section 7, and, finally, we conclude this paper in Section 8.

2 Related Research

Recently, several large open source machine translation projects have been started. Section 3.1 describes the LOGON system, which provides many of the components for our Japanese→English system, Here, we will discuss two other large systems: OpenTrad and OpenLogos.

OpenTrad is a Spanish open source translation initiative consiting of a general MT framework and two engines (Armentano-Oller et al., 2005). The engines are Apertium, a shallow transfer system used for Castillan Spanish↔Catalan, Galician, and Portuguese, with other languages recently added, including English and French. There is also a structural transfer system used for Castillan Spanish↔Basque. Both systems share components (tokeniser, deformatter, reformatter, etc.) and are released under the GPL.

OpenLogos[1] is a 30 year-old commercial transfer system (Scott, 2003) that has recently been released as open source. It can translate from German or English into a number of languages including French, Italian, Spanish, and Portuguese. The system is released under a dual license (commercial/GPL).

Our project is much smaller than either of these, still being closer to its research roots.

3 Japanese→English RBMT with DELPH-IN

The first version of this system is described in detail in Bond et al. (2005). The architecture of our Japanese→English system (hereafter referred to as “Jaen”) is semantic transfer via rewrite rules, as shown in Figure 1. The source text is parsed using an HPSG grammar for the source language, and a semantic analysis in the form of Minimal Recursion Semantics (MRS) is produced. That semantic structure is rewritten using transfer rules into a target-language MRS structure, which is finally used to generate text from a target-language HPSG grammar.

Statistical models are used at various stages in the process. There are seperate models for analyses, transfer and generation, combined as described in Oepen et al. (2007). At each stage we prune the search space, only passing n different results (5 by default) to the next stage.

Although we mainly discuss Jaen in this paper, we have also built a reverse system, Enja, using the same components.

[1] http://logos-os.dfki.de/

3.1 System Components

The grammars and processing systems we use are all being developed within the DELPH-IN [2] project (Deep Linguistic Processing with HPSG Initiative) and are available for download. The lexicon is from an unconnected project (JMdict [3]).

3.1.1 Processing Engines

Jaen uses the LKB (Copestake, 2002) for both parsing and generation. The entire source is released under a very open license, essentially the same as the MIT License. The transfer engine is the MRS rewrite translation engine from the LOGON [4] Norwegian→English MT (Oepen et al., 2004), which is integrated with the LKB.

3.1.2 Grammars

We use HPSG-based grammars of Japanese and English, also from the DELPH-IN project (JACY; Siegel (2000) and the English Resource Grammar (ERG; Flickinger (2000)). Both grammars were originally developed within the Verb*mobil* machine translation effort, but over the past few years have been used for a variety of tasks, including automatic email response and extracting onotlogies from machine readable dictionaries.

The grammars are being developed by seperate groups of researchers, but both are part of the Matrix multilingual grammar engineering effort (Bender et al., 2002). The Matrix consists of a skeleton of grammatical and lexical types, combined with a system of semantic representation known as Minimal Recursion Semantics. The Matrix constitutes a formal backbone for a large scale grammar of, in principle, any language. New grammar resources (e.g., for Italian and Norwegian) were built using the Matrix as a 'starter-kit for grammar writing'. Three existing grammars (English, German, and Japanese) were adapted to the Matrix restrictions.

Other linguistic resources that are available as part of the DELPH-IN open-source repository include a broad-coverage grammar for German and a set of 'emerging' grammars for French, Korean, Modern Greek, Norwegian, Spanish, Swedish, and Portuguese.

[2] http://www.delph-in.net

[3] http://www.csse.monash.edu.au/∼jwb/j_jmdict.html

[4] http://www.emmtee.net

3.1.3 Lexicon

We use JMDict, the Japanese→Multilingual dictionary created by Jim Breen (Breen, 2004) to automatically acquire transfer rules. JMDict has approximately 110,000 main entries, with an additional 12,000 entries for computing and communications technology, and dictionary of over 350,000 proper names. The dictionary is primarily used by non-native speakers of Japanese as an aid to read Japanese. It is widely used, and is increasing in size at the rate of almost 1,000 entries a month (Bond and Breen, 2007).

Because the end users of the dictionary are people, the translations are often more informative than the most common translation equivalents. For example, 医者 *isha* "doctor" is translated as "medical doctor", and フランス語 *furansugo* "French" "French language", in oder to disambiguate them from "Doctor [of Philosophy]" and "French [National]" respectively. These are both correct translations, but they are not necessarily ideal for an MT system: in context, the meaning is normally clear and a translation of just "doctor" or "French" would be preferable.

3.2 Transfer Formalism

MRS (Copestake et al., 2005) is a precise, but underspecified, language-specific semantic representation. MRS structures are flat, unordered collections of elementary predications (EPs) with handles (h) indicating scopal relations, events (e), and entities (x). Figure 2 gives the MRS for the sentence "Research is fun." The sentence is a statement, and the message, `proposition_m_rel(e2)` indicates this. `tanoshii_a_rel(e2,x6)` is an event, and takes `kenkyuu_s_rel(x6)` as its subject. `noun-relation(x6)` nominalizes `kenkyuu_s_rel(x6)`, which is normally an event, turning it into an entity. MRS provides several features that make it attractive as a transfer language, such as uniform representation of pronouns, specifiers, temporal expressions, and the like over grammars. More details can be found in Flickinger et al. (2005).

3.3 Transfer Rules

As illustrated in Oepen et al. (2004), transfer rules take the form of MRS tuples:

```
研究 が 楽しい
 [ LTOP: h1
   INDEX: e2 [ e TENSE: PRES
                MOOD: INDICATIVE
                PROG: - PERF: - ]
   RELS: <
   [ PRED  proposition_m_rel
     LBL: h1
     ARG0: e2
     MARG: h3 ]
   [ PRED  "_kenkyuu_s_rel"
     LBL: h4
     ARG0: x5
     ARG1: u7
     ARG2: u6 ]
   [ PRED  "noun-relation"
     LBL: h8
     ARG0: x5
     ARG1: h9 ]
   [ PRED  proposition_m_rel
     LBL: h9
     ARG0: x5
     MARG: h10 ]
   [ PRED  udef_rel
     LBL: h11
     ARG0: x5
     RSTR: h12
     BODY: h13 ]
   [ PRED  "_tanoshii_a_rel"
     LBL: h14
     ARG0: e2
     ARG1: x5 ] >
   HCONS: < h3 qeq h14, h10 qeq h4,
            h12 qeq h8 > ]
```

Figure 2: MRS for 研究 が 楽しい *research is fun* "kenkyuu ga tanoshii"

```
[CONTEXT:]IN[!FILTER]->OUT
```

where IN(PUT) is rewritten by OUT(PUT), and the optional CONTEXT specifies relations that must be present for the rule to match, and conversely, FILTER specifies relations whose presence blocks a rule from matching. Consider the following transfer rule to translate 言語 *gengo* into "language":

```
gengo-language-mtr :=
[ IN.RELS < [ PRED"_gengo_n_1_rel",
                LBL #h1, ARG0 #x1 ] >,
  OUT.RELS < [ PRED"_language_n_1_rel",
                 LBL #h1, ARG0 #x1 ] > ].
```

This rule rewrites any instance of `gengo_n_1_rel` with `language_n_1_rel`. `#h1` and `#x1` indicate that the LBL and ARG0 arguments of the MRS produced must be preserved. While this may seem like a fairly easy to understand rule, we must repeat the constraint on LBL and ARG0 every time we write a rule to translate nouns. In order to avoid such redundancy in rule writing, LOGON allows the user to specify rule types that can encapsulate common patterns in rules. The above rule can be generalized to cover nouns:

```
noun_mtr := monotonic_mtr &
[IN.RELS < [ LBL #h1, ARG0 #x1 ] >,
 OUT.RELS < [ LBL #h1, ARG0 #x1 ] > ].
```

and our example rule can be rewritten as:

```
gengo-language-mtr := noun_mtr &
[ IN.RELS < [PRED "_gengo_n_1_rel" ] >,
  OUT.RELS <[PRED "_language_n_1_rel"]>].
```

The LOGON system contains a rich definition of rule types - many of which were immediately applicable to Jaen. Jaen inherited from LOGON rule types for open category lexical items such as common nouns, adjectives, and intransitive & transitive verbs. In addition, LOGON contains a number of rule types to specify rules for quantifiers, particles, and conjunctions, providing much of the framework needed to develop Jaen.

3.4 Rule Types Unique to Jaen

Here, we briefly describe a few rule types that were developed to handle linguistic phenomena unique to Japanese→English translation. In Figure 2, we see an example of the Japanese verbal noun, 研究 *kenkyuu* "research" being used as a noun. In Jaen, Japanese verbal nouns are analyzed as events, and they produce messages accordingly. When it is being used as a noun, `kenkyuu_s_rel` is wrapped with the relation noun-relation. We handle these constructions with a special rule that nominalizes the verbal noun by removing its event and the associated message and replacing them with and entity when it appears as a noun:

```
vn-n_jf := monotonic_mtr &
[ CONTEXT.RELS < [ PRED "ja:udef_rel",
                   ARG0 #x0 ] >,
  IN [RELS <[PRED "ja:noun-relation",
          LBL #h6, ARG0 #x0, ARG1 #hp],
         [PRED "ja:proposition_m_rel",
          LBL #hp, ARG0 #ep, MARG #h5 ],
         [PRED #pred, LBL #h0, ARG0 #ep ]>,
      HCONS < qeq & [HARG #h5, LARG #h0 ]>],
  OUT [RELS <[PRED #pred, LBL #h6,
             ARG0 #x0 ]>,
       HCONS < > ] ].
```

In short, this rule type removes the noun-relation and all semantic relations resulting in the

verbal noun's analysis as an event. This change makes it possible to treat verbal nouns identically to regular nouns in the rest of our transfer rules, eliminating the need to create multi-word transfer rules that have to distinguish between nouns and verbal nouns. This simplifies rule development significantly. Thus, a rule to translate 研究 as the noun "research" can now be created using the standard noun template:

```
kenkyuu_s-research_n-omtr := noun_mtr &
[IN.RELS <[PRED "_kenkyuu_s_rel"]>,
 OUT.RELS<[PRED "_research_n_1_rel"]>].
```

4 Expansion of the Core Jaen System

In this section, we describe the process in which the core Jaen system was expanded by targeting a Japanese→English corpus, and using open category transfer rules acquired from a bilingual dictionary to guide the manual development of a small number of transfer rules for the highest occurring closed class rules.

4.1 Targeting the ATR BTEC* Corpus

As development and testing data, we are currently using the ATR Basic Travel Expression Corpus as made available in the IWSLT 2006 evaluation campaign (Paul, 2006). As is indicated in its name, the BTEC* corpus consists of short spoken sentences taken from the travel domain. We selected it because is it a commonly used development set, making our results immediately comparable to a number of different systems, and because our Japanese HPSG parser can successfully analyze approximately 65% of its sentences, providing us with a good base for development. The BTEC* data supplied in the ITWSLT 2006 evaluation campaign consists of almost 40,000 aligned sentence pairs. Sentences average 10.0 words in length for Japanese and 9.2 words in length for English. There are 11,407 unique Japanese tokens and 7,225 unique English tokens.

4.2 Acquiring Open Category Transfer Rules from Bilingual Dictionaries

Nygård et al. (2006) demonstrated that it is possible to learn transfer rules for some open category lexical items using a bilingual Norwegian→English dictionary. They succeeded in acquiring over 6,000 rules for adjectives, nouns, and various combinations thereof. Their method entailed looking up the semantic relations corresponding to words in a translation pair, and matching the results using simple pattern matching to identify compatible rule types.

Our approach is an effort to generalize this approach by using rule templates to generate transfer rules from input source and target MRS structures. Template mappings are used to identify translation pairs where there is a compatible rule type that can be used to create a transfer rule. A template mapping is a tuple consisting of:

- a list of HPSG syntactic categories corresponding to the words in the source translation
- a list of HPSG syntactic categories for the target translation words; and
- the name of the rule template that can be used to construct a transfer rule

Consider the following template mapping:

```
T([noun], [adjective, noun], n-adj+n)
```

This template mapping above identifies a template that creates a rule to translate a Japanese noun into an English adjective-noun sequence.

Transfer rule generation is carried out in the following manner:

1. Look up each word from source-language translation in HPSG lexicon
   - Retrieve syntactic categories and MRS relations
   - Enumerate every possible combination for words with multiple entries
   - Refactor results into separate lists of syntactic categories and MRS relations
2. Repeat 1. for all words in target-language translation
3. Map template mappings onto source and target syntactic categories
   - Translations that match indicate existence of compatible rule template
4. Create a transfer rule by combining the rule template and lists of source and target MRS relations

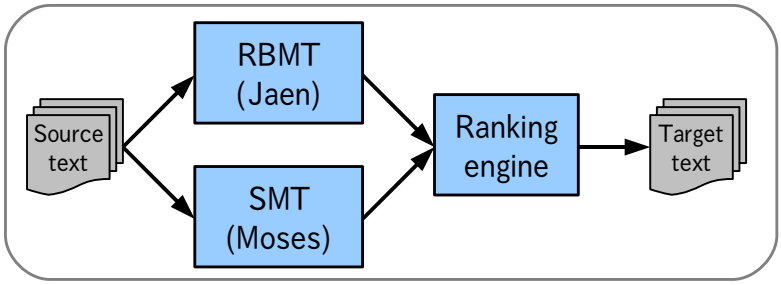


Figure 3: The combined Jaen and Moses system

Using this algorithm we can extract rules from any list of word pairs and have created rules from the EDR[5] Electronic Dictionary, Wikipedia[6] article links, and GIZA++ (Och and Ney, 2003) word alignments from the IWSLT 2006 training data. Our primary source of rules, however, is JMDict. The results of open category transfer rule acquisition from JMDict are summarized in Table 1.

4.2.1 Enhancing the Bilingual Dictionary

The resource bottleneck is a well know problem for machine translation systems. As part of our strategy to overcome it, we are consciously avoiding the creation of specialty lexicons. Instead we are reusing and contributing to an existing dictionary.

JMDict, is an online multilingual Japanese dictionary with a large user base. Users are free to edit and contribute to JMDict, assuring that errors in the lexicon are identified and corrected, and that it can be easily expanded. In order to increase the quality and coverage of JMDict and encourage other users to submit, we make our changes to the dictionary available to the community. In some cases, this means enhancing the descriptive power of JMDict's entries.

We have enhanced the JMdict lexicon in two ways (Bond and Breen, 2007). The first is an explicit distinction between transfer equivalents and explanations:

(1) 点 [てん] ...
```
<gloss g_type="equ">spot</gloss>
<gloss g_type="exp">counter for
goods or items</gloss>
```

The second is to explictly separate disjunctive entries:

(2) 田地 [でんち; でんじ]
```
<gloss>farmland</gloss>
<gloss>rice field or paddy</gloss>
→
<gloss>rice field</gloss>
<gloss>rice paddy</gloss>
```

These two extensions make it possible to produce transfer rules only for those entries which are true translations.

4.3 Handcrafting Closed Category Transfer Rules

In order to decide which semantic relations to write transfer rules for by hand, we used the automatically acquired translation rules in the above section and attempted to translate sentences from the BTEC* corpus. Whenever a relation failed to transfer, the system would be unable to generate a translation, and an error message was produced. We counted the relations and identified the most frequently occurring closed class relations as candidates for handcrafting a transfer rule. There are currently a total of 195 handcrafted rules in our system. A list of the 10 most common untranslatable relations and glosses of the translations we created are given in Table 2.

In handcrafting transfer rules for our system, we also encountered several linguistic problems that needed to be solved in order to achieve high-quality translation results, the most interesting of which was pronoun generation in English. Since our Japanese semantic analyses indicate when arguments of a predicate have been omitted, we came up with a small set of rules that checks what restrictions, if any, are placed on the omitted arguments, and we replace them with underspecified English pronouns, since the nature of the omitted argument is unknown. This leads to over-generation of pronouns, which can cause a com-

[5]http://www2.nict.go.jp/r/r312/EDR/
[6]http://www.wikipedia.org

binatorial explosion in the number of translations for sentences with multiple ellipsed pronouns. To avoid this problem, we only allow pronouns to be inserted for the first two argument slots (roughly corresponding to "subject" and "object").

Other advances made include the treatment of common modal verbs, and natural generation of determiners for negative clauses. We have spent approximately three man months on handcrafting transfer rules.

5 Combining RBMT and SMT

Our end goal is to produce a high-quality, robust machine translation system. To do so, we combine our rule based system with that of an open source statistical machine translation system as shown in Figure 3. The output of the two systems are combined, and a ranking component selects the best possible output. Our current ranking mechanism is a simple cascaded model — we select the RBMT system's output whenever possible, falling back to the SMT system otherwise.

For the fall-back system we use Moses (Koehn et al., 2007), an open source statistical machine translation system that is the result of collaboration at the 2006 John Hopkins University Workshop on Machine Translation. The main component is a beam-search decoder, but it also includes a suite of scripts that, when used together with GIZA++ and SRILM (extensible language modeling toolkit, 2002), make it possible to learn factored phrase-based translation models and carry out end-to-end translation.

We followed the instructions for creating a basic phrase-based factorless system on the Moses homepage[7]. This gave us a system that is comparable to several of that participants in the IWSLT 2006 evaluation.

6 Evaluation

We tracked our coverage on the training set of the IWSLT 2006 evaluation campaign using the rules we acquired and handcrafted as outlined in Section 4.3. Evaluation results are summarized in Table 4. We split all translation pairs into individual sentences by tokenizing on sentence ending punctuation such as "." and "?" yielding a slightly different number of translation sentences than reported in IWSLT 2006's data.

Currently, we have increased our system's coverage tenfold from a starting point of 1.3% up to 13%. In doing so, we are able to translate a large number of sentences with interesting phenomena. Our system's bottleneck is semantic transfer which succeeds over 33% of the time in comparison to the over 65% success rate of parsing and near 60% of generation.

While our currently level of coverage with Jaen makes a quantitative comparison with Moses uninformative, we give a qualitative comparison of the two systems in Figure 3. This small selection of sample translations illustrates the strengths and weaknesses of each of the systems.

As seen in translations 1, 2, and 8, both systems are capable of exactly reproducing the reference for some sentences. Our rule-based system does a better job at preserving structure in translations 4, 5, and 7. Sometimes Moses will omit words entirely; missing the modifier of "hotel" in 4 and the direct object of "see" in 5. While Jaen does not produce perfect translations in these translations, it can be argued that it preserves more of the meaning content of the source sentence.

On the other hand, Jaen often translates quite literally, with the odd-sounding "front money government" being a word-for-word rendering of the Japanese with some slight ambiguity in translating the word corresponding to "government." Sometimes this literal translation can work out well, as in translation 3, where the phrase "this vicinity" is produced in place of the SMT system and reference's use of "here".

Both Jaen and Moses can leave a Japanese word in the translation in-tact. In translation 6, an alignment was not produced for 腹部 *stomach* "fukubu", and it was left untranslated. In translation 2, there is a transliteration of the word 日本 *Japan* "nihon" that is a result of Japanese proper nouns storing transliterations of themselves in their MRS structures. This information is accessible by the English grammar during generation, and, thus "Nihon" is produced.

We feel that the strengths and weaknesses of these two translation systems complement each other; Jaen does a better job at preserving the structure of sentence, where Moses is more ca-

[7] http://www.statmt.org/wmt07/baseline.html

pable at picking up idiomatic, non-compositional translations. Combining their outputs allows us to select the best output possible.

7 Future Work

In addition to the constant work on improving the quality of the system by expanding the inventory of rules, and providing feedback to the component grammars, we are working learning rules from examples. The basic idea is to parse both the source and target and language sentences, then transfer the source and attempt to align the (possibly partial) translation with the parse of the reference translation. Aligned MRS structures can be learned as rules.

A similar approach has been taken by Jellinghaus (2007). The main differences are that they only align very similar sentences; always start the alignment from the root (the handle of the MRS); and directly align the source and target MRSes.

Another area we are working to improve is the translation ranking component of our system combiner. The current method relies on Jaen's statistical models to select the best translation, however, our current models often produce unsatisfiable results. We are exploring methods of directly applying Moses' statistical models to rank system output regardless of its origin.

8 Conclusion

We presented a Japanese→English machine translation system that contains both rule-based and statistical translation engines. All of the components in our system are open source, and excluding the BTEC* data, the resources used in our system are also freely available.

The rule-based translation engine of our system uses a rich semantic representation as a transfer language, allowing the development of powerful transfer rules that produce high-quality translations. By targeting an appropriate corpus for development, automatically acquiring rules from bilingual dictionary, and hand-crafting transfer rules to handle the most common linguistic phenomenon, we were able to greatly extend the RBMT engine's coverage.

The statistical machine translation engine provides a robust fallback for sentences the rule-based system cannot cover. A simple ranking mechanism makes it possible to immediately combine the results of our two translation engine; a better ranking model could help improve overall quality even further.

Comparison of the rule-based and statistical engines showed that their strengths and weaknesses complement each other well. We are optimistic in the potential our combined system has for generating robust and high-quality translations.

Acknowledgments

We would like to thank the members of the LOGON, Hinoki and DELPH-IN projects, especially Stephan Oepen, for their support and encouragement. In addition we would like to thank the developers of the other resources we used in our project, especially JMDict and Moses.

References

Carme Armentano-Oller, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Boyan Bonev, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, and Felipe Sánchez-Martínez. 2005. An open-source shallow-transfer machine translation toolbox: consequences of its release and availability. In *Open-Source Machine Translation: Workshop at MT Summit X*, pages 23–30. Phuket.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14. Taipei, Taiwan.

Francis Bond and James Breen. 2007. Semi-automatic refinement of the JMdict/EDICT Japanese-English dictionary. In *13th Annual Meeting of The Association for Natural Language Processing*, pages 364–367. Kyoto.

Francis Bond, Stephan Oepen, Melanie Siegel, Ann Copestake, and Dan Flickinger. 2005. Open source machine translation with DELPH-IN. In *Open-Source Machine Translation:*

Workshop at MT Summit X, pages 15–22. Phuket.

J. W. Breen. 2004. JMDict: a Japanese-multilingual dictionary. In *Coling 2004 Workshop on Multilingual Linguistic Resources*, pages 71–78. Geneva.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics. An introduction. *Research on Language and Computation*, 3(4):281–332.

SRILM An extensible language modeling toolkit. 2002. Andreas stolcke. In *International Conference on Spoken Language Processing*, volume 2, pages 901–904. Denver.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28. (Special Issue on Efficient Processing with HPSG).

Dan Flickinger, Jan Tore Lønning, Helge Dyvik, Stephan Oepen, and Francis Bond. 2005. SEM-I rational MT: Enriching deep grammars with a semantic interface for scalable machine translation. In *Machine Translation Summit X*, pages 165–172. Phuket.

Michael Jellinghaus. 2007. *Automatic Acquisition of Semantic Transfer Rules for Machine Translation*. Master's thesis, Universität des Saarlandes.

Philipp Koehn, Wade Shen, Marcello Federico, Nicola Bertoldi, Chris Callison-Burch, Brooke Cowan, Chris Dyer, Hieu Hoang, Ondrej Bojar, Richard Zens, Alexandra Constantin, Evan Herbst, Christine Moran, and Alexandra Birch. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 177–180. Prague. URL `http://www.statmt.org/moses/`.

Lars Nygård, Jan Tore Lønning, Torbjørn Nordgård, and Stephan Oepen. 2006. Using a bi-lingual dictionary in lexical transfer. In *EAMT-2006*, pages 233–238. Oslo.

Franz Josef Och. 2005. *Statistical Machine Translation: Foundations and Recent Advances*. MT Summit, Phuket.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Stephan Oepen, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, and Victoria Rosèn. 2004. Som å kapp-ete med trollet? Towards MRS-based Norwegian–English Machine Translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*. Baltimore, MD.

Stephan Oepen, Erik Velldal, Jan Tore Løning, Paul Meurer, and Victoria Rosen. 2007. Towards hybrid quality-oriented machine translation —On linguistics and probabilities in MT—. In *Eleventh International Conference on Theoretical and Methodological Issues in Machine Translation: TMI-2007*. Skövde. (this volume).

Michael Paul. 2006. Overview of the IWSLT 2006 Evaluation Campaign. In *Proc. of the International Workshop on Spoken Language Translation*, pages 1–15. Kyoto, Japan.

Bernard (Bud) Scott. 2003. The Logos model: An historical perspective. *Machine Translation*, 18:1–72.

Melanie Siegel. 2000. HPSG analysis of Japanese. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 265 – 280. Springer, Berlin, Germany.

Tatsuya Sukehiro, Mihoko Kitamura, and Toshiki Murata. 2001. Collaborative translation environment 'Yakushite.Net'. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium: NLPRS-2001*, pages 769–770. Tokyo.

Andy Way. 1999. A hybrid architecture for robust MT using LFG-DOP. *Journal of Experimental and Theoretical Artificial Intelligence*, 11. Special Issue on Memory-Based Language Processing.

Rule type	BTEC* vocabulary	Total rules	Examples
Adj→Verb	98	250	不安→to worry
Verb→Adj	239	268	有り得る→likely
Adj+Noun→Adj+Noun	478	527	渋い ワイン→white wine
Intransitive Verb	1,273	2,519	現れる→to appear
Noun→Adj.+Noun	2,262	2,787	悪玉→bad character
Adj, Adverb	2,660	3,023	青い→green
Noun+Noun→Noun	2,945	3,135	アイディア 商品→novelty
Noun→Noun+Noun	2,100	3,588	甘党→sweet tooth
Noun+Noun→Adj+Noun	3,974	4,482	暗黒 物質→dark matter
Transitive Verb	3,299	5,344	選ぶ→ to choose
Noun+Noun→Noun+Noun	5,303	7,909	操り 芝居→puppet show
Noun	14,489	16,242	字→character
Total	39,120	50,074	

Table 1: Results of automatic transfer rule acquisition from JMDict

Frequency	Semantic relation	Translation
25,927	"_ni_p_rel"	に → in, to, into
25,056	"cop_id_rel"	だ, です → to be
22,976	"_no_p_rel"	XのY → X Y, X's Y, Y of X
10,375	"_de_p_rel"	で → in, on, at, with
9,696	"rareru_rel"	∼られる → passive
9,528	"neg_v_rel"	∼ない → negation
8,848	"_exist_v_rel"	ある → to be, to have
7,627	"_kono_q_rel"	この → this
4,173	"tai_rel"	∼たい → to want to
3,588	"_hour_n_rel"	時 → time, hour

Table 2: Most frequently occuring source language relations and their hand-crafted translations

	Jaen	Moses	Reference
1	Are Japanese dogs big?	It is a big dog in Japan?	Are Japanese dogs big?
2	Where is there a Nihon embassy?	Where is the Japanese Embassy?	Where is the Japanese Embassy?
3	Is there a hotel in this vicinity?	Is there a hotel near here?	Is there a hotel around here?
4	A center hotel.	The hotel.	The Center Hotel.
5	Did you see criminals?	Did you see the?	Did you see who did it?
6	Abdomens hurt.	腹部 aches.	I have a stomach ache.
7	Please do an allergy check.	I am allergic to check, please.	I'd like to have an allergy test, please.
8	Is it a front money government?	Do I need to pay in advance?	Do I need to pay in advance?

Table 3: Sample translations from Jaen and Moses systems

IWSLT 2006 Training data results				
Parsing	28,175	/	42,699	65.98%
Transfer	9,355	/	28,175	33.20%
Generation	5,523	/	9,355	59.04%
Overall	**5,523**	/	**42,699**	**12.93**%

Table 4: Coverage for Jaen on the IWSLT 2006 traning data

Towards Hybrid Quality-Oriented Machine Translation

— On Linguistics and Probabilities in MT —

Stephan Oepen[♣♠], Erik Velldal[♣], Jan Tore Lønning[♣], Paul Meurer[♡], Victoria Rosén[♢], and Dan Flickinger[♠]

[♣] Department of Informatics, University of Oslo, Norway
[♠] Center for the Study of Language and Information, Stanford University, USA
[♡] Centre of Culture, Language and Information Technology, University of Bergen, Norway
[♢] Department of Linguistics, University of Bergen, Norway

Abstract

We present a hybrid MT architecture, combining state-of-the-art linguistic processing with advanced stochastic techniques. Grounded in a theoretical reflection on the division of labor between rule-based and probabilistic elements in the MT task, we summarize per-component approaches to ranking, including empirical results when evaluated in isolation. Combining component-internal scores and a number of additional sources of (probabilistic) information, we explore discriminative re-ranking of n-best lists of candidate translations through an eclectic combination of knowledge sources, and provide evaluation results for various configurations.

1 Background—Motivation

Machine Translation is back in fashion, with data-driven approaches and specifically Statistical MT (SMT) as the predominant paradigm—both in terms of scientific interest and evaluation results in MT competitions. But (fully-automated) machine translation remains a hard—if not ultimately impossible—challenge. The task encompasses not only all strata of linguistic description—phonology to discourse—but in the general case requires potentially unlimited knowledge about the actual world and situated language use (Kay, 1980, 1997). Although the majority of commercial MT systems still have large sets of hand-crafted rules at their core (often using techniques first invented in the 1960s and 1970s), MT research in the once mainstream linguistic tradition has become the privilege of a small, faithful minority.

Like a growing number of colleagues, we question the long-term value of *purely* statistical (or data-driven) approaches, both practically and scientifically. Large (parallel) training corpora remain scarce for most languages, and word- and phrase-level alignment continue to be active research topics. Assuming sufficient training material, statistical translation quality still leaves much to be desired; and probabilistic NLP experience in general suggests that one must expect 'ceiling' effects on system evolution. Statistical MT research has yet to find a satisfactory role for linguistic analysis; on its own, it does not further our understanding of language.

Progress on combining rule-based and data-driven approaches to MT will depend on a sustained stream of state-of-the-art, MT-oriented linguistics research. The Norwegian LOGON initiative capitalizes on linguistic precision for high-quality translation and, accordingly, puts scalable, general-purpose linguistic resources—complemented with advanced stochastic components—at its core. Despite frequent cycles of overly high hopes and subsequent disillusionment, MT in our view is the type of application that may demand knowledge-heavy, 'deep' approaches to NLP for its ultimate, long-term success. Much like Riezler & Maxwell III (2006) and Llitjós & Vogel (2007)—being faithful minority members ourselves—we approach a hybrid MT architecture with a semantic transfer backbone as our vantage point. Plurality of approaches to grammatical description, reusability of component parts, and the interplay of linguistic and stochastic processes are among the strong points of the LOGON system.

In the following, we provide a brief overview of the LOGON architecture (§ 2) and a bit of theoretical reflection on the role of probability theory

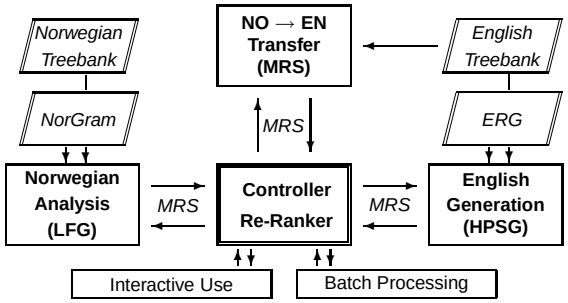


Figure 1: Schematic system architecture: the central controller brokers intermediate representations among the three processing components, accumulating candidate translations and, ultimately re-ranking the n-best list.

in finding optimal translations (§ 3). Sections § 4 through § 6 review component-internal ranking in the LOGON pipeline. Finally, § 7 outlines our approach to end-to-end re-ranking, including empirical results for various setups. We conclude with reflections on accomplishments so far and ongoing work in § 8.

2 LOGON—Hybrid Deep MT

The LOGON consortium—the Norwegian universities of Oslo (coordinator), Bergen, and Trondheim—has assembled a 'deep' MT prototype over the past four years, expending around fifteen person years on its core translation system. The LOGON pipeline comprises grammar-based parsing, transfer of underspecified Minimal Recursion Semantics (MRS; Copestake, Flickinger, Pollard, & Sag, 2005), and full tactical generation (aka realization). NorGram, the analysis grammar, is couched in the LFG framework and has been continuously developed at the University of Bergen since 1999. Conversely, the generation grammar, ERG (Flickinger, 2000), builds on the HPSG theory of grammar, and has been under development at CSLI Stanford since around 1993. While both analysis and generation deploy general-purpose linguistic resources and processing tools, LOGON had to develop its MRS transfer formalism and Norwegian – English (NoEn) transfer grammar from scratch. The transfer engine—unification-based, resource-sensitive rewriting of MRS terms—constitutes a new generic tool (that is already used for other language pairs and even non-MT tasks), but most of the NoEn transfer grammar is specific to the LOGON language pair and application. Figure 1 shows a schematic view of the LOGON architecture; Oepen et al. (2004) provide a more detailed overview of the LOGON approach.

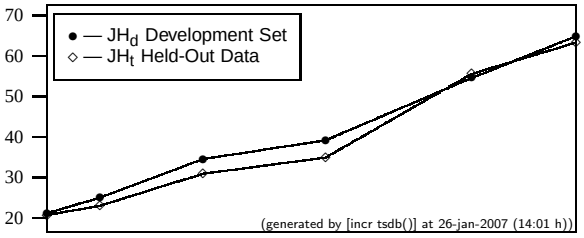
set	#	words	coverage	strings
JH$_d$	2146	12.6	64.8	266
JH$_t$	182	11.7	63.2	114.6

Table 1: LOGON development and held-out corpora (for the *Jotunheimen* segment). Average string length and end-to-end coverage on the two sets are comparable, but the average number of candidate translations is higher on the development data.

In a nutshell, the role of the rule-based components in LOGON is to delineate the space of grammatically and semantically coherent translations, while the ranking of competing hypotheses and ultimately the selection of the best candidate(s) is viewed as a probabilistic task. Parsing, transfer, and realization each produce, on average, a few hundred candidate outputs for one input. Hence, exhausting the complete fan-out combinatorics can be prohibitively expensive, and typically we limit the number of hypotheses passed downstream to a relatively small n-best list. For all results reported presently, the fan-out branching factor was limited to a maximum of five output candidates from parsing and (within each branch) transfer; because there is no further downstream processing after generation, we can afford more candidate realizations per input MRS—for a total of up to $5 \times 5 \times 50 = 1250$ distinct fan-out outcomes. However, it is quite common for distinct fan-out paths to arrive at equivalent outputs, for example where the same modifier attachment ambiguity may be present in the source and target language.

Both our linguistic resources, search algorithms, and statistical models draw from contemporary, state-of-the art techniques and ongoing research in larger, non-MT communities. In this regard, the LOGON demonstrator provides a novel blending of approaches, where the majority of its component parts and linguistic resources have independent value (and often are used in parallel in other research efforts and applications).

The consortium circumscribed its domain and ambitions by virtue of a reference corpus of around 50,000 words of running text, six published tourism booklets on back-country activities



17-oct-2005 (20:08 h) – 18-jan-2007 (03:07 h)

Figure 2: Evolution of end-to-end coverage over time: percentage of *Jotunheimen* inputs with at least one translation.

in Norway. In addition to one original translation, we contracted up to two additional reference translations; about ten per cent of the parallel corpus was held out for evaluation. Table 1 summarizes core metrics of the training and test sections of the *Jotunheimen* booklets, the largest segment and the one for which three reference translations are available. For model training and evaluation, about 670 of the Norwegian inputs and all (∼6,000) English references were manually treebanked (see below).

Aiming primarily to gauge the utility of its 'pure' setup (rather than for a complete MT solution) at the current stage, the consortium did not 'diffuse' its linguistic backbone with additional robustness measures. Accordingly, the overall error rate is the product of per-component errors, and gradually building up end-to-end coverage—specifically harmonizing semantics for a wide variety of constructions cross-linguistically—was a major part of system development. Figure 2 depicts the evolution of end-to-end coverage in the past year and a half. Upon completion of active development, system performance on held-out data was determined retroactively (for earlier versions). In terms of end-to-end coverage at least, it is reassuring to observe that there are few differences between system behavior on development vs. held-out data: for this domain and genre, the final LOGON demonstrator translates about two thirds of its inputs.

3 Some Theoretical Reflections

Given our transfer system, where each of the three steps fan out, there are several possibilities for adding a stochastic component. What should be maximized, and how?

The first possibility is to rank the different components sequentially, one at a time. First rank the

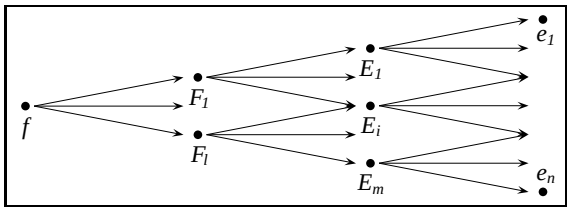


Figure 3: Abstract fan-out tree: each processing component operates non-deterministically, and distinct inputs can, in principle, give rise to equivalent outputs.

results of parsing and choose the topmost candidate, call it F_1. Then consider all the results of invoking transfer on F_1, and choose the one ranked highest, E_1. And finally choose the highest ranked realization e_1 of E_1. We will refer to this output as the *first translation*, corresponding to the top branch in Figure 3.

The second possibility is to try to find the *most likely path* through the fan-out tree, i.e. try to maximize:

$$\arg\max_{i,j,k} P(e_k|E_j)P(E_j|F_i)P(F_i|f)$$

The two approaches do not always yield the same result. Take as an example a sentence f with two different analyses, F_1 and F_2, where the main difference between the two is that a particular word is ambiguous between a noun reading in F_1, and a verb reading in F_2. If the noun has many alternative realizations in the target language while the verb has few, the most likely path might be one that chooses the verb, i.e. passes through F_2.

The third possibility for the end-to-end ranking is to try to find the *most likely translation*, i.e.

$$\arg\max_{e} \sum_{F_i} \sum_{E_j} P(e_k|E_j)P(E_j|F_i)P(F_i|f)$$

This might result in a different top-ranked candidate than the most likely path in cases where several different paths result in the same output. Considering PP attachment ambiguities, for example, distinct intermediate semantic representations (pairs of E_is and F_js) can yield the same target string.

Which concept should we try to model? From a theoretical point of view, there are good arguments for choosing what we have called the first translation. It makes sense to try to select the most likely interpretation of what the producer of

the source sentence has intended independently of how it gets translated. If one instead selects the most likely path, or the most likely translation, one might select a less likely interpretation of what the speaker had intended.

Our argument for the *first translation* can be illustrated within our earlier example of a word-level noun vs. verb ambiguity in analysis. The many different realizations of the noun in the target language may fall into classes of near synonyms, in which case it does not matter for the quality of the result which synonym is chosen. Even though each of the individual realizations has a low probability, it may be a good translation.

Observe here also that an automatic evaluation measure—measuring the similarities to a set of reference translations, like the BLEU metric (Papineni, Roukos, Ward, & Zhu, 2002)—will favor the view of *most likely translation*. We conjecture, however, that a human evaluation will correspond better to the first translation.

From a theoretical point of view, it seems most correct to go for the first translation. But it presupposes that we choose the correct interpretation of the source sentence, which we cannot expect to always do. In cases where we have chosen an incorrect analysis, this might be revealed by trying to translate it into the target language and consider the result. If all the candidate translations sound bad—or have a very low probability—in the target language, that can be evidence for dispreferring this analysis. Hence information about probabilities from later components in the pipeline may be relevant, not for overwriting analysis probabilities, but for helping in selecting them.

We will in the following first review how LOGON employs component ranking for choosing the first translation, and then consider an end-to-end re-ranking which attempts to find the most probable translation, by directly estimating the posterior translation probability $P(e|f)$.

4 Parse Selection

In a sister project to LOGON, the TREPIL project, a toolkit for building parsebanks of LFG analyses is being developed (Rosén, Smedt, & Meurer, 2006). This toolkit, called the LFG Parsebanker, was used to build a treebank for the LOGON development corpus. Parse selection in LOGON uses training data from this treebank; all sentences with full parses with low ambiguity (fewer than 100 readings) were at least partially disambiguated.

ambiguity	#	exact match	five-best
$50-100$	16	34.4 (17.2)	56.2 (55.0)
$25-49$	28	30.4 (21.4)	62.5 (54.3)
$10-24$	43	58.1 (25.3)	89.5 (73.9)
$2-9$	53	70.8 (35.1)	96.2 (91.0)
total	**140**	**53.8 (27.3)**	**84.3 (74.3)**
$50-100$	16	43.7 (17.2)	81.2 (55.0)
$25-49$	28	50.0 (21.4)	78.6 (54.3)
$10-24$	43	67.4 (25.3)	90.7 (73.9)
$2-9$	53	72.6 (35.1)	100. (91.0)
total	**140**	**63.2 (27.3)**	**90.7 (74.3)**

Table 2: Evaluation of parse selection with a model trained with standard feature function templates of the XLE (upper part, as used in LOGON,) and with a discriminant model (lower part, not yet used). Figures are given for the percentage of exact matches and matches among the five top-ranked analyses. Figures in parentheses show a random choice baseline. Both models were trained on seven of nine treebanked texts and evaluated on the two remaining texts.

The parse selection method employed in the LOGON demonstrator uses the stochastic disambiguation scheme and training software developed at PARC (Riezler & Vasserman, 2004). The XLE system provides a set of parameterized feature function templates that must be expanded in accordance with the grammar or the training set at hand. Application of these feature functions to the training data yields feature forests for both the labeled data (the partially disambiguated parse forests) and the unlabeled data (the full parse forests). These feature forests are the input to the statistical estimation algorithm, which generates a property weights file that is used to rank solutions.

One of the challenges in applying the probability model to a given grammar and training set is the choice of appropriate feature functions. We have pursued two approaches for choosing feature functions. In the first approach, we started with a significant subset of the predefined feature function templates and expanded each of them in all possible ways that would result in a non-zero value on at least one parse in the train-

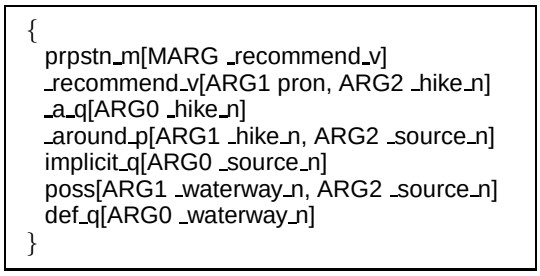


Figure 4: Variable-free reduction of the MRS for the utterance 'We recommend a hike around the waterway's sources'.

ing set; this could be done automatically. The second approach is motivated by the hypothesis that discriminants, as used in manual annotation (Carter, 1997), represent promising alternative feature functions to the predefined templates. Initial tests (see table 2) show that the discriminant approach (which is not yet used in the LOGON system) scores better than the template-based approach.

5 Ranking Transfer Outputs

While MRS formulae are highly structured graphs, Oepen & Lønning (2006) suggest a reduction into a variable-free form that resembles elementary dependency structures. For the ranking of transfer outputs, MRSs are broken down into basic dependency triples, whose probabilities are estimated by adaptation of standard n-gram sequence modeling techniques. The actual training is done using the freely available CMU SLM toolkit (Clarkson & Rosenfeld, 1997).

Based on a training set of some 8,500 in-domain MRSs, viz. the treebanked version of the English translations of the (full) LOGON development corpus, our target language 'semantic model' is defined as a smoothed tri-gram model over the reduction of MRSs into dependency triples. Figure 4 shows an example structure, corresponding to a total of ten triples, including $\langle$_around_p, ARG1, _hike_n$\rangle$. The 'vocabulary' of the model comprises some 4,400 distinct semantic predicates and role labels, for a total number of around 51,000 distinct triples. Similarly, post-transfer English MRSs are broken down into segments of dependency triples and ranked according to the perplexity scores assigned by the semantic model.

We lack a transfer-level 'treebank' to evaluate MRS ranking in isolation, but in lieu of such data, we can contrast end-to-end system performance on the JH$_t$ test set. When passing an unranked, random selection of five transfer outputs downstream, the success rate in generation drops to 82.7 per cent (down from 86.5 per cent in ranked, five-best mode). Restricting the comparison to the 109 items that translate in both configurations, our BLEU score over the *first* translation drops from 37.41 to 30.29.[1]

6 Realization Ranking

Realization ranking is the term we use for the task of discriminating between multiple surface forms generated for a given input semantics. By adapting methods previously used for parse selection, we are able to use treebank data for training a discriminative log-linear model for the conditional probability of a surface realization given an input MRS. Traditionally, however, the standard approach to tackling this problem of indeterminacy in generation is to use an *n-gram language model* (Langkilde & Knight, 1998; White, 2004; inter alios). Candidate strings are then ranked according to their 'fluency', indicated by the probabilities assigned by the LM. As a baseline for our discriminative model, we trained a tri-gram language model on an unannotated version of the British National Corpus (BNC), containing roughly 100 million words. As in the case of the MRS ranker, we used the CMU SLM toolkit for training, resulting in a Witten-Bell discounted back-off model.

When evaluated in terms of exact match accuracy on the JH$_d$ development set,[2] the LM ranker achieves 53.2%, which is well above the random choice baseline of 28.7%. However there are many well-known limitations inherent to the n-gram approach, such as its inability to capture long-range dependencies and dependencies between non-contiguous words. More generally, the simple n-gram models are purely surface ori-

[1] BLEU measures in all our experiments are calculated using the freely available NIST toolkit (in its version 11b).

[2] Note that, when evaluating realization rankers in isolation, we use a different version of the JH$_d$ data set. The MRSs in the generation treebank are here always underspecified with respect to information structure, such as passivization and topicalization. This means that the level of indeterminacy is somewhat higher than what is typically the case within the LOGON MT setting.

model	exact match	five-best	WA
BNC LM	53.24	78.81	0.882
Log-Linear	72.28	84.59	0.927

Table 3: Performance of the realization rankers. *BNC LM* is the n-gram ranker trained on the raw text version of the BNC. *Log-Linear* shows 10-fold cross-validated results for the discriminative model trained on a generation treebank, including the LM scores as a separate feature.

ented and thereby fail to capture dependencies that show a structural rather than sequential regularity. All in all, there are good reasons to expect to devise better realization rankers by using models with access to grammatical structure. Velldal, Oepen, & Flickinger (2004) introduced the notion of a *generation treebank*, which facilities the training of discriminative log-linear models for realization ranking in a similar fashion as for parse disambiguation. For further background on log-linear models, see § 7.

Our discriminative realization ranker uses a range of features defined over the derivation trees of the HPSG linguistic sign, recording information about local sub-tree configurations, vertical dominance relations, n-grams of lexical types, and more (Velldal & Oepen, 2006). When trained and tested by ten-fold cross-validation on a generation treebank created for the JH_d data set, this model achieves 70.28% exact match accuracy, clearly outperforming the n-gram-based LM by a good margin (again, the random choice baseline is 28.7%). However, by including the scores of the LM as an additional feature, we are able to further boost accuracy up to 72.28%. Table 3 summarizes the results of the two different types of realization rankers. The evaluation also includes exact match accuracy within the five top-ranked candidates, as well as average sentence-level *word accuracy* (WA), which is a string similarity measure based on edit distance.

7 End-to-End Re-Ranking

Section § 3 already suggests one consideration in favor of re-ranking the complete list of candidate translations once fan-out is complete: component-internal probabilistic models are fallible. Furthermore, besides analysis-, transfer-, and realization-internal information, there are additional properties of each hypothesized pair $\langle f, e \rangle$ that can be brought to bear in choosing the 'best' translation, for example a measure of how much reordering has occurred among corresponding elements in the source and target language, or the degree of harmony between the string lengths of the source and target.

Log-linear models provide a very flexible framework for discriminative modeling that allows us to combine disparate and overlapping sources of information in a single model without running the risk of making unwarranted independence assumptions. In this section we describe a model that directly estimates the posterior translation probability $P_\lambda(e|f)$, for a given source sentence f and translation e. Although the re-ranker we describe here is built on top of a hybrid baseline system, the overall approach is similar to that described by Och & Ney (2002) in the context of SMT.

Log-Linear Models A log-linear model is given in terms of (a) a set of *specified features* that describe properties of the data, and (b) an associated set of *learned weights* that determine the contribution of each feature. One advantage of working with a discriminative re-ranking setup is that the model can use global features that the baseline system would not be able to incorporate. The information that the feature functions record can be arbitrarily complex, and a given feature can even itself be a separate statistical model. In the following we first give a brief high-level presentation of conditional log-linear modeling, and then we go on to present the actual feature functions in our setup.

Given a set of m real-valued features, each pair of source sentence f and target sentence e are represented as a feature vector $\Phi(f, e) \in \Re^m$. A vector of weights $\lambda \in \Re^m$ is then fitted to optimize some objective function of the training data. For the experiments reported in this paper the weights are fitted to maximize the conditional (or *pseudo*) likelihood (Johnson, Geman, Canon, Chi, & Riezler, 1999).[3] In other words, for each input source sentence in the training data we seek to maximize

[3] For estimation we use the TADM open-source toolkit (Malouf, 2002), using its *limited-memory variable metric* as the optimization method. As is standard practice, the model is regularized by including a zero-mean Gaussian prior on the feature weights to reduce the risk of overfitting.

the probability of its annotated reference translation relative to the other competing candidates. However, for future work we plan to also experiment with optimizing the scores of a given evaluation metric (e.g. BLEU) directly, following the Minimum Error Rate approach of Och (2003).

The three most fundamental features that are supplied in our log-linear re-ranker correspond to the three ranking modules of the baseline system, as described in Sections § 4, § 5, and § 6 above. In other words, these features record the scores of the parse ranker, the MRS ranker, and the realization ranker, respectively. But our re-ranker also includes several other features that are not part of the baseline model.

Other Features Our experiments so far have taken into account another eight properties of the translation process, in some cases observing internal features of individual components, in others aiming to capture global information. The following paragraphs provide an informal overview of these additional features in our log-linear re-ranking model.

LEXICAL PROBABILITIES One additional feature type in the log-linear model corresponds to *lexical translation probabilities*. These are estimated on the basis of a small corpus of Norwegian – English parallel texts, comprising 22,356 pairs of aligned sentences.[4] First, GIZA^{++} is used for producing word alignments in both directions, i.e. using both languages as source and target in turn. On the basis of these alignments we then estimate a maximum likelihood translation table, again in both directions.[5] Finally, for each bi-directional sentence pair $\langle e, f \rangle$ and $\langle f, e \rangle$, the corresponding feature in the end-to-end ranker is computed as the length-normalized product of all pairwise word-to-word probabilities.

STRING PROBABILITY Although a part of the (conditional) realization ranker already, we include the string probability (according to the tri-gram language model trained on the BNC) of candidate translations e_k as an independent indicator of output fluency.

DISTORTION Elementary predications (EPs) in our MRS are linked to corresponding surface elements, i.e. sub-string pointers. Surface links are preserved in transfer, such that post-generation, for each EP—or group of EPs, as transfer need not be a one-to-one mapping—there is information about its original vs. its output sub-string span. To gauge reordering among constituents, for both the generator input and output, each EP is compared pairwise to other EPs in the same MRS, and each pair classified with regard to their relative surface positions. Comparing the input and output MRS, we consider corresponding pairs of EP pairs; the distortion metric for a pair of aligned EPs measures their class difference, where for example a change from overlapping to adjacent is penalized mildly, while inverting a precedence relation comes at a higher cost. Finally, the distortion metric for a pair of MRSs is the sum of their per-EP distortion metrics, normalized by the total number of EP pairs.

STRING HARMONY Seeing typological similarity between Norwegian and English, much like for the distortion metric, we assume that there are systematic correspondences at the string level between the source and its translation. To enable the re-ranker to take into account length effects, we include the ratio of word counts, $|e|/|f|$, as a feature in the model.

TRANSFER METRICS Two additional features capture information about the transfer step: the total number of transfer rules that were invoked (as a measure of transfer granularity, e.g. where idiomatic transfer of a larger cluster of EPs contrasts with stepwise transfer of component EPs), as well as the ratio of EP counts, $|E|/|F|$.

SEMANTIC DISTANCE Generation proceeds in two phases: a chart-based bottom-up search enumerates candidate realizations, of which a final semantic compatiblity test selects the one(s) whose MRS is subsumed by the original generator input MRS (Carroll & Oepen, 2005). Given an imperfect input (or error in the generation grammar), it is possible for none of the candidate outputs to fulfill the semantic compatiblity test. In this case, the generator will gradually relax MRS com-

[4] Of these, 9,410 sentences are taken from the LOGON development data, while an additional 12,946 sentences are from the English-Norwegian Parallel Corpus (Oksefjell, 1999).

[5] The ML estimation of the lexical probabilities, as well as the final word alignments produced from the output of GIZA^{++}, are carried out using the training scripts provided by Phillip Koehn, and as distributed with the phrase-based SMT module Pharaoh (Koehn, 2004).

parison, going through seven pre-defined levels of semantic mismatch, which we encode as one integer-valued feature in the re-ranking model.

Training the Model While batch translating, the LOGON controller records all candidate translations, intermediate semantic representations, and a large number of processing and resource consumption properties in a database, which we call a *profile* (in analogy to software engineering; Oepen et al., 2005). Given the system configuration summarized in Sections § 2 through § 6, we use the JH_d batch profile to train and optimize a log-linear re-ranker. The experimentation infrastructure, here, is essentially the same as in our discriminative realization ranker—the combination of the [incr tsdb()] profiler, the TADM maximum entropy toolkit, and tools for efficient cross-valiation experiments with large data and feature sets (Velldal, 2007).

For training purposes, we mechanically 'annotated' candidate translations by means of the sentence-level NEVA string similarity measure, applied to actual LOGON outputs compared to JH_d reference translations. NEVA is a reformulation of BLEU that avoids many of the problems associated with applying BLEU at the sentence level, and is computed as the arithmetic mean of the raw n-gram precision scores (Forsbom, 2003). For each source sentence, we mark the translation(s) with maximum NEVA score (among all candidate outputs for this input) as preferred, thus constructing an empirical distribution where estimation of log-linear model parameters amounts to adjusting conditional probabilities towards higher NEVA scores.

Seeing that the model includes diverse feature types—probabilities, perplexity values, un-normalized log-linear scores, and non-probabilistic quantities—feature values are normalized into a comparable range, using min-max scaling. The hyper-parameters of the model—the TADM convergence threshold and variance of the Gaussian prior—were optimized by ten-fold cross-validation on the training corpus.

Empirical Results Table 4 summarizes end-to-end system performance, measured in BLEU scores, for various strategies of selecting among the n-best lists obtained from $5 \times 5 \times 50$ fan-out. In all cases, scoring has been reduced to those inputs actually translated by the LOGON system, i.e. 64.8% and 63.2% of the development (JH_d) and held-out (JH_t) corpora, respectively. As a baseline measure, we used random choice of one output in each context (averaged over twenty iterations), resulting in (estimable) BLEU scores of 34.18 and 30.84, respectively.

set	#	chance	first	LL	top	judge
$\textbf{JH}_\textbf{d}$	1391	34.18	40.95	44.10	49.89	–
$\textbf{JH}_\textbf{t}$	115	30.84	35.67	38.92	45.74	46.32

Table 4: BLEU scores for various re-ranking configurations, computed over only those cases actually translated by LOGON (second column). For all configurations, BLEU results on the training corpus are higher by about four points.

As an upper bound on re-ranking efficacy, Table 4 provides two 'oracle' scores: the first, labeled *top*, is obtained from selecting translations with maximal NEVA scores, i.e. using sentence-level NEVA as a proxy for corpus-level BLEU. The second, labeled *judge*, reflects the annotations of a human judge on the JH_t held-out data: considering all available candidates, a native speaker of (American) English and near-native speaker of Norwegian, in each case, picked the translation judged most appropriate (or, in some cases, least awful). Oracle BLEU scores reach 49.89 and 46.32, for JH_d and JH_t, respectively.

Finally, the column labeled *first* in Table 4 corresponds to the *first translation* concept introduced in § 3 above, and the *LL* column to our log-linear re-ranker (maximizing the *log-likelihood* of the training data). Both clearly improve over the random choice baseline, but the re-ranker outperforms the first translation approach by a large margin—thus returning on the investment of extra fan-out and end-to-end re-ranking. However, at BLEU scores of 44.10 and 38.92, respectively, our current re-ranking setup also leaves ample room for further improvements towards the 'oracle' upper bound. We anticipate that fine-tuning the log-linear model, inclusion of additional features, and experimentation with different estimation techniques (see below) will allow us to narrow this differential further.

8 Conclusions—Outlook

The future of MT has been (mis-)diagnosed as 'just around the corner' since the beginning of time, and there is no basis to expect a breakthrough in fully-automated MT in the foreseeable future. But yet we see progress along the way, specifically in the sustained development of large-scale, general-purpose language technology and its ever tighter integration with refined stochastic techniques.

Among the main results of the Norwegian LOGON initiative is its proof-of-concept demonstrator for quality-oriented, hybrid MT grounded in independently developed computational grammars. The tight coupling of hand-built linguistic resources results in an MT pipeline where, to a very high degree, all candidate translations are (a) related to the source utterance in a systematic—albeit at times unlikely—way and (b) grammatically well-formed. Combining an n-best beam search through the space of fan-out combinatorics with stochastic rankers at each step, as well as with discriminative end-to-end re-ranking yields a flexible solution, offering a clear precision vs. efficiency trade-off. For its bounded domain (and limited vocabulary of around 5,000 lexemes), the LOGON system succeeds in translating about two thirds of unseen running text, where BLEU scores and project-internal inspection of results suggest a high degree of output quality. This configuration could, in principle, be an interesting value proposition by itself—as a tool to professional translators, for example. A more systematic, human judgment study of system outputs (for various selection strategies) is currently underway, and we expect results to become available in June this year.

In ongoing work, we aim to further improve re-ranking performance, for example by assessing the relative contribution of individual features, fine-tuning parameter estimation, and including additional properties. Our current maximum likelihood training of the log-linear model is based on a binarized empirical distribution, where for each input we consider the candidate translation(s) with maximum NEVA score(s) as preferred, and all others as dis-preferred. Obviously, however, the degradation in quality among alternate candidates is continuous (rather than absolute), and we have started experimentation with a graded empirical distribution, adapting the approach of Osborne (2000) to the re-ranking task. Finally, in a parallel refinement cycle, we aim to contrast our current (LL) re-ranking model with Minimum Error Rate (MER) training, a method that aims to estimate model parameters to directly optimize BLEU scores (or another quality metric) as its objective function.

Trading coverage for increased output quality may be economic for a range of tasks—say as a complement to other tools in the workbench of a professional translator. Our re-ranking approach, with access to rich intermediate representations, probabilities, and confidence measures, provides a fertile environment for experimentation on *confidence-centric* MT. Applying thresholding techniques on the probability distribution of the re-ranking model, for example, we plan to experimentally determine how much translation quality can be gained by making the candidate selection more restrictive. Alternatively, one can imagine applying yet another model to this task, a classifier deciding on which candidate translations constitute worthy outputs, and which are best suppressed.

The availability of off-the-shelf SMT tools has greatly contributed to re-energized interest and progress in MT in the recent past. We believe that advances in hybrid MT would equally benefit from a repository of general-purpose, easy-to-use linguistic resources. Except for the proprietary XLE, all LOGON results—treebanks, grammars, and software—are available for public download.

References

Carroll, J., & Oepen, S. (2005). High-efficiency realization for a wide-coverage unification grammar. In R. Dale & K. F. Wong (Eds.), *Proceedings of the 2nd International Joint Conference on Natural Language Processing* (Vol. 3651, pp. 165 – 176). Jeju, Korea: Springer.

Carter, D. (1997). The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering.* Madrid, Spain.

Clarkson, P., & Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge Toolkit. In *Proceedings of EuroSpeech.* Rhodes, Greece.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal Recursion Semantics. An introduction.

Journal of Research on Language and Computation, 3(4), 281 – 332.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering, 6 (1)*, 15 – 28.

Forsbom, E. (2003). Training a super model look-alike: Featuring edit distance, n-gram occurrence, and one reference translation. In *Proceedings of the workshop on machine translation evaluation: Towards systemizing MT evaluation, held in conjunction with MT SUMMIT IX*. New Orleans, USA.

Johnson, M., Geman, S., Canon, S., Chi, Z., & Riezler, S. (1999). Estimators for stochastic 'unification-based' grammars. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics* (pp. 535 – 541). College Park, MD.

Kay, M. (1980). *The proper place of men and machines in translation* (Technical Report # CSL-80-11). Palo Alto, CA: Xerox Palo Alto Research Center.

Kay, M. (1997). It's still the proper place. *Machine Translation, 12*(1 - 2), 35 – 38.

Koehn, P. (2004). Pharaoh. A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas* (pp. 115 – 124). Washington DC.

Langkilde, I., & Knight, K. (1998). The practical value of n-grams in generation. In *Proceedings of the 9th International Workshop on Natural Language Generation* (pp. 248 – 255). Ontario, Canada.

Llitjós, A. F., & Vogel, S. (2007). A walk on the other side. Adding statistical components to a transfer-based translation system. In *Proceedings of the HLT-NAACL workshop on Syntax and Structure in Statistical Translation.* Rochester, NY.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning.* Taipei, Taiwan.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics* (pp. 160 – 167). Sapporo, Japan.

Och, F. J., & Ney, H. (2002). Discriminative training and Maximum Entropy models for statistical machine translation. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics* (pp. 295 – 302). Philadelphia, PA.

Oepen, S., Dyvik, H., Flickinger, D., Lønning, J. T., Meurer, P., & Rosén, V. (2005). Holistic regression testing for high-quality MT. Some methodological and technological reflections. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation.* Budapest, Hungary.

Oepen, S., Dyvik, H., Lønning, J. T., Velldal, E., Beermann, D., Carroll, J., Flickinger, D., Hellan, L., Johannessen, J. B., Meurer, P., Nordgård, T., & Rosén, V. (2004). Som å kapp-ete med trollet? Towards MRS-based Norwegian – English Machine Translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation.* Baltimore, MD.

Oepen, S., & Lønning, J. T. (2006). Discriminant-based MRS banking. In *Proceedings of the 4th International Conference on Language Resources and Evaluation.* Genoa, Italy.

Oksefjell, S. (1999). A description of the English-Norwegian Parallel Corpus. Compilation and further developments. *International Journal of Corpus Linguistics*, *4*(2), 197 – 219.

Osborne, M. (2000). Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the 18th International Conference on Computational Linguistics.* Saarbrücken, Germany.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002). BLEU. A method for automatic evaluation of machine translation. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics* (pp. 311 – 318). Philadelphia, PA.

Riezler, S., & Maxwell III, J. T. (2006). Grammatical machine translation. In *Proceedings of the Human Language Technology Conference and Annual Meeting of the North American Association for Computational Linguistics* (pp. 248 – 255). New York, NY.

Riezler, S., & Vasserman, A. (2004). Incremental feature selection and l_1 regularization for relaxed maximum-entropy modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Barcelona, Spain.

Rosén, V., Smedt, K. D., & Meurer, P. (2006). Towards a toolkit linking treebanking and grammar development. In *Proceedings of the 5th Workshop on Treebanks and Linguistic Theories* (pp. 55 – 66). Prague, Czech Republic.

Velldal, E. (2007). *Stochastic realization ranking.* Doctoral dissertation, University of Oslo, Oslo, Norway. (in preparation)

Velldal, E., & Oepen, S. (2006). Statistical ranking in tactical generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Sydney, Australia.

Velldal, E., Oepen, S., & Flickinger, D. (2004). Paraphrasing treebanks for stochastic realization ranking. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories* (pp. 149 – 160). Tübingen, Germany.

White, M. (2004). Reining in CCG chart realization. In *Proceedings of the 3rd International Conference on Natural Language Generation.* Hampshire, UK.

Reducing Human Assessment of Machine Translation Quality to Binary Classifiers

Michael Paul[†‡] **and Andrew Finch**[†‡] **and Eiichiro Sumita**[†‡]
† NICT Spoken Language Communication Group
‡ ATR Spoken Language Communication Research Labs
Hikaridai 2-2-2, Keihanna Science City, 619-0288 Kyoto
{Michael.Paul,Andrew.Finch,Eiichiro.Sumita}@nict.go.jp

Abstract

This paper presents a method to predict human assessments of machine translation (MT) quality based on the combination of binary classifiers using a *coding matrix*. The multiclass categorization problem is reduced to a set of binary problems that are solved using standard classification learning algorithms trained on the results of multiple automatic evaluation metrics. Experimental results using a large-scale human-annotated evaluation corpus show that the decomposition into binary classifiers achieves higher classification accuracies than the multiclass categorization problem. In addition, the proposed method achieves a higher correlation with human judgments on the sentence-level compared to standard automatic evaluation measures.

1 Introduction

The evaluation of MT quality by humans is cost- and time-intensive. Various automatic evaluation measures have been proposed to make evaluations of MT outputs cheaper and faster. Recent evaluation campaigns on newswire[1] and travel data[2] investigated how well these evaluation metrics correlate with human judgments. The results showed that high correlations to human judges were obtained for some metrics when ranking MT system outputs on the document-level. However, each automatic metric focuses on different aspects of the translation output and its correlation towards human judges depends on the type of human assessment (for example *fluency* or *adequacy*). Moreover, none of the automatic metrics turned out to be satisfactory in predicting the translation quality of a single translation.

This paper presents a method to predict human assessments of machine translation (MT) quality based on the combination of binary classifiers. The multiclass categorization problem is reduced to a set of binary problems that are solved using standard classification learning algorithms. Binary classifiers are trained on features of multiple automatic evaluation metrics, such as BLEU and METEOR. The learned discriminative models are applied sentence-wise to MT outputs producing binary indicators of translation quality on the sentence-level. The multiclass classification problem is then solved by combining the results of the binary classifiers using a *coding matrix*.

The human and automatic evaluation metrics investigated in this paper are described in Section 2. Section 3 gives a brief overview on related research on predicting human assessments and outlines the main differences to the proposed method. Section 4 outlines the

[1] NIST MT evaluations, http://www.nist.gov/speech/tests/mt

[2] IWSLT evaluations, http://www.slc.atr.jp/IWSLT2006

Table 1: Human Assessment

fluency	
5	Flawless English
4	Good English
3	Non-native English
2	Disfluent English
1	Incomprehensible

adequacy	
5	All Information
4	Most Information
3	Much Information
2	Little Information
1	None

acceptability	
5	Perfect Translation
4	Good Translation
3	Fair Translation
2	Acceptable Translation
1	Nonsense

proposed method. The framework of reducing multiclass to binary classification and the combination of the binary results to solve the multiclass classification problem are described in detail. The effectiveness of the proposed method is evaluated in Section 5 for English translations of Chinese and Japanese source sentences in the travel domain.

2 Assessment of Translation Quality

Various approaches on how to assess the quality of a translation have been proposed. In this paper, human assessments of translation quality with respect to the *fluency*, the *adequacy* and the *acceptability* of the translation are investigated. *Fluency* indicates how natural the evaluation segment sounds to a native speaker of English. For *adequacy*, the evaluator was presented with the source language input as well as a "gold standard" translation and has to judge how much of the information from the original translation is expressed in the translation (White et al., 1994). *Acceptability* judges how easy-to-understand the translation is (Sumita et al., 1999). The *fluency*, *adequacy* and *acceptability* judgments consist of one of the grades listed in Table 1.

The high cost of such human evaluation metrics has triggered a huge interest in the development of automatic evaluation metrics for machine translation. Table 2 introduces some metrics that are widely used in the MT research community.

Table 2: Automatic Evaluation Metrics

BLEU:	the geometric mean of n-gram precision of the system output with respect to reference translations. Scores range between 0 (worst) and 1 (best) (Papineni et al., 2002)
NIST:	a variant of BLEU using the arithmetic mean of weighted n-gram precision values. Scores are positive with 0 being the worst possible (Doddington, 2002)
METEOR:	calculates unigram overlaps between a translation and reference texts using various levels of matches (*exact*, *stem*, *synonym*). Scores range between 0 (worst) and 1 (best) (Banerjee and Lavie, 2005)
GTM:	measures the similarity between texts by using a unigram-based F-measure. Scores range between 0 (worst) and 1 (best) (Turian et al., 2003)
WER:	*Word Error Rate*: the minimal edit distance between the system output and the closest reference translation divided by the number of words in the reference. Scores are positive with 0 being the best possible (Niessen et al., 2000)
PER:	*Position independent WER*: a variant of WER that disregards word ordering (Och and Ney, 2001)
TER:	*Translation Edit Rate*: a variant of WER that allows phrasal shifts (Snover et al., 2006)

3 Prediction of Human Assessments

Most of the previously proposed approaches to predict human assessments of translation quality utilize supervised learning methods like *decision trees* (DT), *support vector machines* (SVM), or *perceptrons* to learn discriminative models that are able to come closer to human quality judgments. Such classifiers can be trained on a set of features extracted from human-evaluated MT system outputs.

The work described in (Quirk, 2004) uses statistical measures to estimate confidence on the word/phrase level and gathers system-specific features about the translation process itself to train binary classifiers. Empirical thresholds on automatic evaluation scores are utilized to distinguish between good and bad translations. He also investigates the feasabil-

ity of various learning approaches for the multiclass classification problem for a very small data set in the domain of technical documentation.

(Akiba et al., 2001) utilized DT classifiers trained on multiple *edit-distance* features where combinations of lexical (stem, word, part-of-speech) and semantic (thesausus-based semantic class) matches were used to compare MT system outputs with reference translations and to approximate human scores of *acceptability* directly.

(Kulesza and Shieber, 2004) trained a binary SVM classifier based on automatic scoring features in order to distinguish between "human-produced" and "machine-generated" translations of newswire data instead of predicting human judgments directly.

The approach proposed in this paper also utilizes a supervised learning method to predict human assessments of translation quality, but differs in the following two aspects:

(1) *Reduction of Classification Perplexity*: The decomposition of a multiclass classification task into a set of binary classification problems reduces the complexity of the learning task resulting in higher classification accuracy.

(2) *Feature Set*: Classifiers are trained on the results of multiple automatic evaluation metrics (see Table 2) thus taking into account different aspects of translation quality addressed by each of the metrics. The method does not depend on a specific MT system nor on the target language. It can be applied without modification to any translation or target language as long as reference translations are available.

4 Human Assessment Prediction based on Binary Classifier Combination

The proposed prediction method is divided into three phases: (1) a *learning phase* in which binary classifiers are trained on the feature set that is extracted from a database of human and machine-evaluated MT system outputs, (2) a *decomposition phase* in which the optimal set of binary classifiers that maximizes the classification accuracy of the recombination step on a development set is selected, (3) an *application phase* in which the binary classifiers are applied to unseen sentences, and the results of the binary classifiers are combined using the optimized coding matrix to predict a human score.

4.1 Learning Phase

Discriminative models for the multiclass and binary classification problem are obtained by using standard learning algorithms. The proposed method is not limited to a specific classification learning method. For the experiments described in Section 5, we utilized a standard implementation of decision trees (Rulequest, 2004).

The feature set consists of the scores of the seven automatic evaluation metrics listed in Table 2. All automatic evaluation metrics were applied to the input data sets consisting of English MT outputs whose translation quality was manually assessed by humans using the metrics introduced in Section 2. In addition to the metric scores, metric-internal features, like *ngram-precision* scores, *length ratios* between references and MT outputs, etc. were also utilized, resulting in a total of 54 training features.

4.2 Decomposition Phase

There are many ways in which a multiclass problem can be decomposed into a number of binary classification problems. The most well-known approaches are the *one-against-all* and *all-pairs*. In the *one-against-all* approach, a classifier for each of the classes is trained where all training examples that belong to that class are used as positive examples and all others as negative examples. In the *all-pairs* approach, classifiers are trained for each pair of classes whereby all training examples that do not belong to any of the classes in question are ignored (Hastie and Tibshirani, 1998).

Such decompositions of the multiclass problem can be represented by a *coding matrix* $\mathcal{M}$

where each class c of the multiclass problem is associate with a row of binary classifiers b. If k is the number of classes and l is the number of binary classification problems, the coding matrix is defined as:

$$\mathcal{M} = (\ m_{i,j}\ )_{i=1,\ldots,k;j=1,\ldots,l}$$

$$m_{i,j} \in \{-1,\ 0,\ +1\},$$

where k is the number of classes and l is the number of binary classification problems. If the training examples that belong to class c are considered as positive examples for a binary classifier b, then $m_{c,b}$=+1. Similarily, if $m_{c,b}$=–1 the training examples of class c are used as negative examples for the training of b. $m_{c,b}$=0 indicates that the respective training examples are not used for the training of classifier b (Dietterich and Bakiri, 1995; Allwein et al., 2000). Examples of coding matrices for *one-against-all* and *all-pairs* (k=3, l=3) are given in Table 3.

Table 3: Coding Matrix Examples

one-against-all

	$c_1 \bullet c_{23}$	$c_2 \bullet c_{13}$	$c_3 \bullet c_{12}$
c_1	+1	–1	–1
c_2	–1	+1	–1
c_3	–1	–1	+1

all-pairs

	$c_1 \bullet c_2$	$c_1 \bullet c_3$	$c_2 \bullet c_3$
c_1	+1	+1	0
c_2	–1	0	+1
c_3	0	–1	–1

For the experiments described in Section 5, we utilized both *one-against-all* and *all-pairs* binary classifiers. In addition, *boundary* classifiers were trained on the whole training set. In this case, all training examples annotated with a class better than the class in question were used as positive examples and all other training examples as negative examples. Table 4 lists the 17 binary classification problems that were utilized to decompose the human assessment problems introduced in Section 2.

In order to identify the optimal coding matrix for the respective tasks, the binary classifiers were first ordered according to their classification accuracy on the development set. In the second step, the multiclass performance was evaluated iteratively, where the worst performing binary classifier was omitted from the coding matrix after each iteration. Finally, the coding matrix achieving the best classification accuracy for the multiclass task was used for the evaluation of the test set. The optimized coding matrix reflects the standard bias-variance trade-off balancing the discriminative power and the reliability of the binary classifier combination.

Table 4: Decomposition of Human Assessment of Translaton Quality

type	binary classifier
one-against-all	5, 4, 3, 2, 1
all-pairs	5_4, 5_3, 5_2, 5_1, 4_3, 4_2, 4_1, 3_2, 3_1, 2_1
boundary	54_321, 543_21

4.3 Application Phase

Given an input example, all binary classifiers are applied once for each column of the coding matrix resulting in a vector v of l binary classification results. The multiclass label is predicted as the label c for which the respective row r of $\mathcal{M}$ is "closest".

In (Allwein et al., 2000), the distance between r and v, is calculated by (a) a generalized *Hamming distance* that counts the number of positions for which the corresponding vectors are different and (b) a *loss-based decoding* that takes into account the magnitude of the binary classifier scores. For the experiments described in Section 5, we adopted the Hamming-distance approach.

An example for the distance calculation is given in Table 5. Lets assume that the application of the three binary classifiers listed in Table 3 results in the classification vector v = (+1, +1, –1) for a given input. Using the *one-against-all* coding matrix, the minimal distance for v is 1 for both matrix rows, c_1 and c_2. In case of a draw, the priority order of binary classifiers obtained on the development set is used to identify the more reliable row. For the *all-pairs* coding matrix, class c_1 would be selected due to its lesser distance.

Table 5: Coding Matrix Application

$v = (+1, +1, -1)$

type	multiclass	distance	selection
one-against-all	c_1	1	c_1 or c_2
	c_2	1	
	c_3	3	
all-pairs	c_1	1	c_1
	c_2	3	
	c_3	2	

5 Evaluation

The evaluation of the proposed method was carried out using the *Basic Travel Expression Corpus* (BTEC). This contains tourism-related sentences similar to those usually found in phrase books for tourists going abroad (Kikui et al., 2003). In total, 3,524 Japanese input sentences were translated by MT systems of various types[3] producing 82,406 English translations. 54,576 translations were annotated with human scores for *acceptability* and 36,302 translations were annotated with human scores for *adequacy*/*fluency*. The distribution of the human scores for the given translations is summarized in Figure 1. In case multiple human judgments were assigned to a single translation output, the median of the respective human scores was used in our experiments.

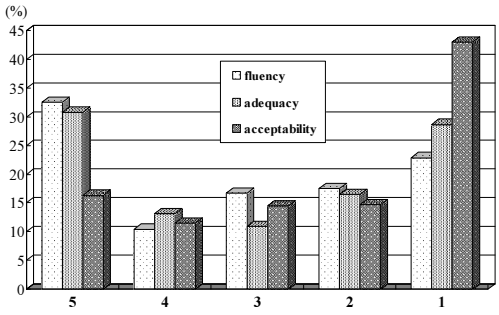


Figure 1: Human Score Distribution

The annotated corpus was split into three data sets: (1) the *training set* consisting of 25,988 translations for *adequacy*/*fluency* and 49,516 MT outputs for *acceptability*, (2) the *development set* consisted of 2,024 sentences (4 MT outputs for each of 506 input sentences) for all three metrics, and (3) the *test set* taken from the IWSLT evaluation campaign (CSTAR03 data set, 506 input sentences). For *fluency* and *adequacy*, 7,590 test sentences with 15 MT outputs for each were available. For *acceptability*, 3,036 sentences with 6 MT outputs for each were used for evaluation.

[3]Most of the translations were generated by statistical MT engines, but 5 example-based and 5 rule-based MT systems were also utilized. These engines were state-of-the-art MT engines. Some participated in the IWSLT evaluation campaign series and some were in-house MT engines.

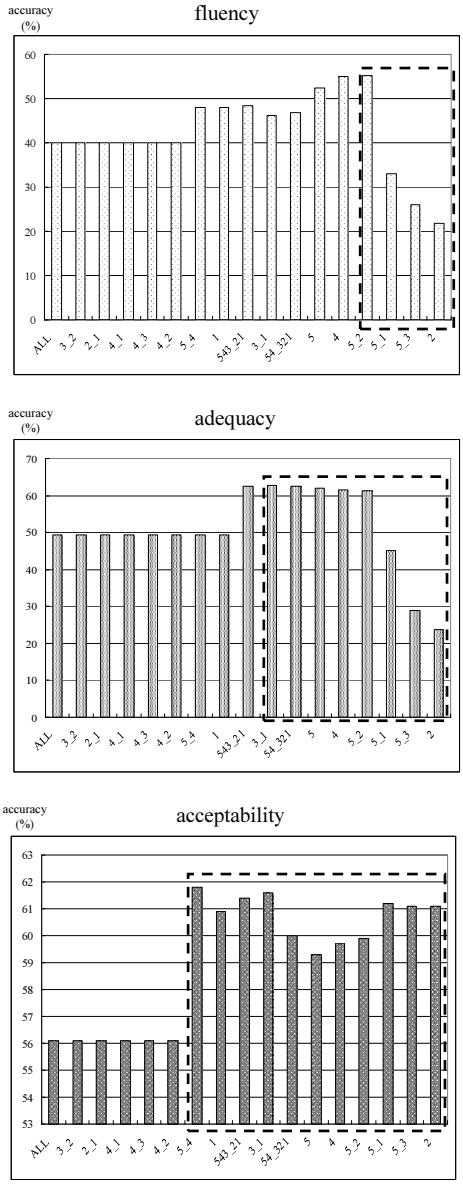


Figure 2: Coding Matrix Optimization

5.1 Coding Matrix Optimization

Figure 2 summarizes the iterative evaluation of the binary classification combination us-

ing the development set as described in Section 4.2. Starting with the complete coding matrix (*ALL*), the worst performing binary classification is omitted in the next iteration. The dashed square indicates the subset of binary classifiers selected for the coding matrix utilized for the test set evaluation.

5.2 Classification Accuracy

The baseline of the multiclass classification task was defined as the class most frequently occuring in the training data set. Table 6 summarizes the baseline performance for all three subjective evaluation metrics.

Table 6: Baseline Accuracy

fluency	*adequacy*	*acceptability*
32.5%	**30.8%**	**43.0%**

The classification accuracies of the multiclass task, i.e. the multiclass classifier learned directly from the training set, and the binary classifier performance is summarized in Figure 3. The results show that the learning approach outperforms the baseline of the multiclass classification task for all three metrics gaining 16.7% for *fluency*, 26.8% for *adequacy* and 18.1% for *acceptability*.

Moreover, the performance of the binary classifiers varies widely, depending on the classification task as well as the evaluation metric. Accuracies of 80%-90% were achieved for the *all-against-one* classifiers, 75%-81% for the *boundary* classifiers, and 55%-91% for the *all-pairs* classifiers.

The proposed method combines the binary classifiers according to the optimized coding-matrix. The results are shown in Figure 4. The classification accuracy of the proposed method is 55.2% for *fluency*, 62.6% for *adequacy* and 62.3% for *acceptability*. Thus, the proposed method outperforms the baseline as well as the multiclass classification task for all subjective evaluation metrics achieving a gain of 22.7% / 6.0% in *fluency*, 31.5% / 6.6% in *adequacy* and 19.3% / 1.2% in *acceptability* compared to the baseline / multiclass performance, respectively.

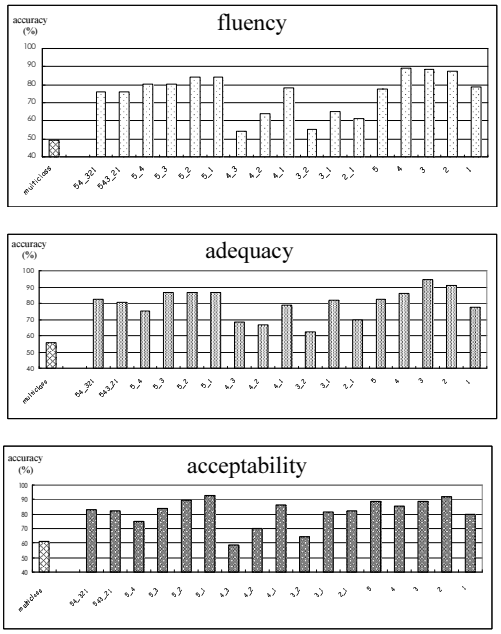


Figure 3: Classifier Accuracy

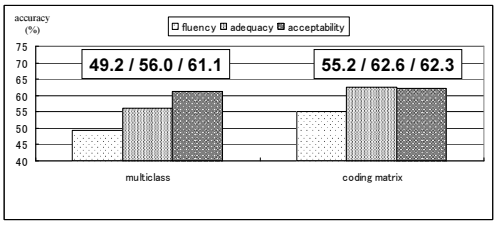


Figure 4: Classifier Combination Accuracy

5.3 Correlation to Human Assessments

In order to investigate the correlation of the proposed metrics towards human judgments on the sentence-level, we calculated the Spearman rank correlation coefficient for the obtained results. In addition, we used the multiclass classifier and the automatic evaluation metrics listed in Table 2 to rank the test sentences and calculate its Spearman rank correlation towards human assessments. The correlation coefficients are summarized in Figure 5.

The results show that the proposed method outperforms all other metrics achieving correlation coefficients of 0.632 / 0.759 / 0.769 for *fluency* / *adequacy* / *acceptability*, respectively. Concerning the automatic evaluation metrics, METEOR achieved the high-

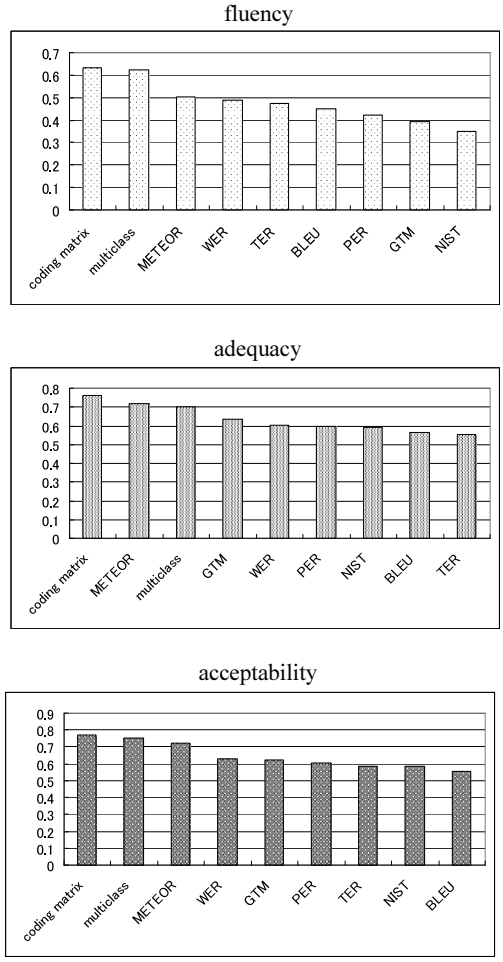


Figure 5: Correlation with Human Assessments

est correlation towards human assessment on sentence-level for all three subjective evaluation metrics. The correlation of the remaining automatic metrics is considerably lower and depends largely on the type of human assessment.

5.4 Upper Bound

In order to get an idea about the potential of the proposed method, we simluated the upper bound of the method by randomly adjusting the prediction result of each binary classifier to achieve a certain classification accuracy and applied the coding matrix approach to the set of binary classifiers having the same classification accuracy. Figure 6 shows the upper boundary of the proposed method for classification accuracies between 60% and 100% whereby the respective optimized coding matrix of the experiments described in Section 5.2 were used for *fluency*, *adequacy* and *acceptability*, respectively. The *all_binary* result shows the performance when the baseline coding matrix using all 17 binary classifiers is applied.

The results show that for each metrics the multiclass classification task performance is almost linearly related to the performance of the binary classifiers and that improving the accuracy of the binary classifiers will result in a better overall performance.

Two potential improvements of the proposed method, that we would like to investigate in the near future, are (1) additional features that help to classify the given task more acurately, and (2) the automatic learning of the optimal combination of binary classifiers with respect to the overall system performance.

6 Conclusion

In this paper, we proposed a robust and reliable method to learn discriminative models based on the results of multiple automatic evaluation metrics to predict translation quality at the sentence level. The prediction is carried out by reducing the multiclass classification problem to a set of binary classification tasks and combining the respective results using a coding matrix in order to predict the multiclass label for a given input sentence.

The effectiveness of the proposed method was verified using three types of human assessment of translation quality commonly used within the MT research community. The experiments showed that the proposed method outperforms a baseline method that selects the most frequent class contained in the training set and a standard multiclass classification model (decision tree) that learns its discriminative model directly from the training corpus. The proposed method achieved a gain of 22.7% / 6.0% in *fluency*, 31.5% / 6.6% in *adequacy* and 19.3% / 1.2% in *acceptability* compared to the baseline / multiclass performance, respectively. Moreover, the proposed metric achieved high correlation to human judgments

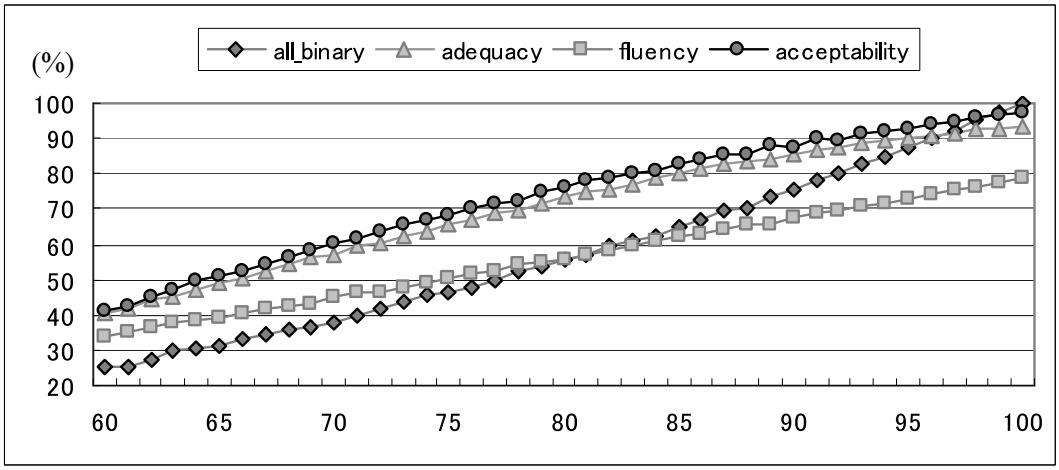


Figure 6: Upper Boundary of Reducing Multiclass to Binary Classifier

at the sentence-level outperforming not only the multiclass approach, but also all of the automatic scoring metrics utilized.

Future extensions of the proposed method will investigate the use of additional features, such as the confidence estimation features proposed in (Blatz et al., 2003) or the recently proposed source language features for MT evaluation in (Liu and Gildea, 2007). We would expect this to improve the performance of the binary classifiers and boost the overall performance further.

References

Yasuhiro Akiba, Kenji Imamura, and Eiichiro Sumita. 2001. Using multiple edit distances to automatically rank machine translation output. In *Proc. of MT Summit VIII*, pages 15–20.

Erin Allwein, Robert Schapire, and Yoram Singer. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence estimation for statistical machine translation. In *Final Report of the JHU Summer Workshop*.

Thomas G. Dietterich and Ghulum Bakiri. 1995. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. of the HLT 2002*, pages 257–258, San Diego, USA.

Trevor Hastie and Robert Tibshirani. 1998. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471.

Genichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto. 2003. Creating corpora for speech-to-speech translation. In *Proc. of the EUROSPEECH03*, pages 381–384, Geneve, Switzerland.

Alex Kulesza and Stuart M. Shieber. 2004. A learning approach to improving sentence-level MT evaluation. In *Proc. of the TMI04*, USA.

Ding Liu and Daniel Gildea. 2007. Source-language features and maximum correlation training for machine translation evaluation. In *Proc. of the NAACL-HLT*, pages 41–48, Rochester NY, USA.

Sonja Niessen, Franz J. Och, Gregor Leusch, and Hermann Ney. 2000. An evaluation tool for machine translation: Fast evaluation for machine translation research. In *Proc. of the 2nd LREC*, pages 39–45, Athens, Greece.

Franz J. Och and Hermann Ney. 2001. Statistical multi-source translation. In *Proc. of the MT Summit VIII*, pages 253–258, Santiago de Compostella, Spain.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the 40th ACL*, pages 311–318, Philadelphia, USA.

Christopher B. Quirk. 2004. Training a sentence-level machine translation confidence measure. In *Proc. of 4th LREC*, pages 825–828, Portugal.

Rulequest. 2004. Data mining tool c5.0. http://rulequest.com/see5-info.html.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of the AMTA*, pages 223–231, Cambridge and USA.

Eiichiro Sumita, Setsuo Yamada, Kazuhide Yamamoto, Michael Paul, Hideki Kashioka, Kai Ishikawa, and Satoshi Shirai. 1999. Solutions to problems inherent in spoken-language translation: The ATR-MATRIX approach. In *Proc. of the MT Summit VII*, pages 229–235, Singapore.

Joseph Turian, Luke Shen, and I. Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proc. of the MT Summit IX*, pages 386–393, New Orleans, USA.

John White, Theresa O'Connell, and Francis O'Mara. 1994. The ARPA MT evaluation methodologies: evolution, lessons, and future approaches. In *Proc of the AMTA*, pages 193–205.

Sub-Phrasal Matching and Structural Templates in Example-Based MT

Aaron B. Phillips
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
aphillips@cmu.edu

Abstract

In this work I look at two different paradigms of Example-Based Machine Translation (EBMT). I combine the strengths of these two systems and build a new EBMT engine that combines sub-phrasal matching with structural templates. This synthesis results in higher translation quality and more graceful degradation, yielding 1.5% to 7.5% relative improvement in BLEU scores.

1 Introduction

Example-Based Machine Translation (EBMT) introduced the notion of phrasal translation that has subsequently been championed by Phrasal Statistical Machine Translation (PSMT). Exact phrasal translations are usually highly accurate and retain the nuances of the text. However, unless one focuses exclusively on a (very) small domain, it is unreasonable to assume that a corpus will provide exact phrasal translations of everything one wants to translate. Thus, methods of backing off and synthetically generating translations based on "similar" examples are increasingly important. In this work I introduce a new EBMT Engine named Cunei[1] (Construction of Unknown Examples by Induction) that combines two different paradigms of EBMT: sub-phrasal matching and structural templates. The goal of this work is to provide highly accurate translation when possible, but also allow for more graceful degradation through a form of structural generalization.

2 Overview

The EBMT system at CMU, Panlite (Brown, 1996), is shallow in the sense that it only indexes lexical tokens. It performs well primarily because it is capable of indexing very large corpora and efficiently extracting exact lexical translations. When an example covering the full input sentence is not present in the corpus, Panlite attempts to match any sub-part of the sentence. This is done by matching all possible token sequences without any respect for phrasal boundaries. The retrieved examples are placed in a lattice that is subsequently decoded by a language modeler. This particular EBMT system is actually very similar to PSMT as it consists of a phrase extraction phase followed by a language modeler that performs phrase selection and reordering. The main differences lie in the details of the calculations and the fact that Panlite does not attempt to retain a true probabilistic model.

Not all EBMT implementations take this approach. In particular, Gaijin (Veale and Way, 1997) retrieves examples from a corpus based on their structural similarity. The marker hypothesis stipulates that a closed set of words in every language can be used to identify the syntactic structure of a sentence. These markers are typically conjunctions, prepositions, determiners, and quantifiers. Gaijin employs the marker hypothesis to segment sentences into constituent phrases as shown in Figure 1. Each constituent phrase is headed by a marker that represents the type of that constituent. The particular sequence of constituent phrases describes the structure of the sentence.

[1] Named after Cuneiform, the oldest writing system to be translated.

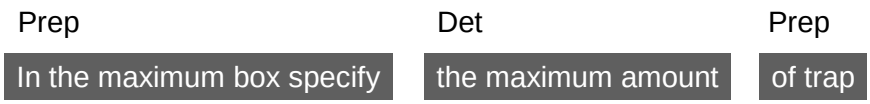


Figure 1. Sentence Segmented by Marker Hypothesis (Veale and Way, 1997)

This structure, rather than a language model, dictates phrasal selection and reordering. An example from the corpus that has the same sequence of constituent phrases becomes the master template for translation. When this template has lexical mismatches with the sentence to be translated, "grafting" is used to replace an entire phrasal constituent with another (more similar) phrasal constituent found in the corpus. Likewise, if particular words within a phrasal constituent do not match the input, "keyhole surgery" is performed to substitute individual lexical items. For either type of substitution to be performed, the structure (part-of-speech tag or head-of-phrase marker) must be equivalent.

Both of these EBMT systems build a final translation by synthetically combining together smaller units of translation. In the case of Panlite, the units are any sequence of lexical tokens, and they are combined together using a language modeler. On the other hand, the units in the Gaijin system are constituents identified by the marker hypothesis, and they are combined together by a single structural template from the corpus that matches the entire sentence.

Cunei attempts to bring together the strengths of Panlite and Gaijin. This new system maintains the indexing scheme and sub-phrasal matching found in Panlite and adds to this a "light" version of the structural matching found in the Gaijin system. Instead of using constituent phrases identified by the marker hypothesis as the structure of each sentence (Figure 1), Cunei uses only the sequence of part-of-speech tags as shown in Figure 2. Gaijin was built for a relatively small corpus and as such it was necessary to use a more general structure. The sequence of part-of-speech tags is very specific, but by leveraging a large corpus I expect to find many structural examples. This system will not, however, require one template to translate the entire sentence, but rather, like Panlite, will find examples corresponding to any sub-section of the input sentence. Cunei passes the resulting lattice to the same language modeler used by Panlite for decoding.

Using part-of-speech tags to form structural templates is similar to the Transfer (Xfer) approach described in (Carbonell et al., 2002) and (Probst et al., 2003). The structural templates in Cunei are, in some respects, more limited as they do not incorporate morphological features. However, the role of the structural templates in Cunei is different as they are merely a backoff mechanism to be used when an exact lexical match is not present, and thus, generality is desired. In addition, the structural templates in Cunei are entirely data-driven. Instead of using a lexicon that specifies words available for substitution, Cunei fills the structural template using phrases present in the lattice that have the same part-of-speech sequence. The scores associated with each phrase in the lattice are taken into account when constructing a new example from the structural template.

Cunei was developed and evaluated translating text from Arabic to English. I expected the difference in word order between these two languages to work well with structural templates. However, the system is language-neutral and could easily be applied to any language pair for which part-of-speech taggers and parallel text are available.

3 Building Cunei

3.1 Preprocessing

For structural matching, it was important to process the English and Arabic in the same format as the Penn Treebank because this was expected by the part-of-speech taggers I used. A handful of regular expressions were applied to re-format the text and perform some simple cleanup. Next, I

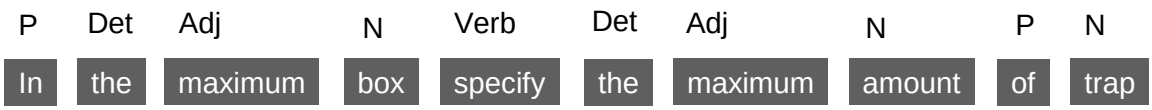


Figure 2. A "Lite" Structure: Sentence with Part-Of-Speech Tags

used MXPOST (Ratnaparkhi, 1996) to apply part-of-speech tags to the English text and ASVMTools (Diab et al, 2004) to perform segmentation and part-of-speech tagging on the Arabic text. It is worthwhile to point out that because of the two different part-of-speech taggers, the naming conventions for the tags were not always the same. This does not make a difference to Cunei as there are no a priori rules that assume a noun should replace a noun. Rather, substitutions are determined at run-time based on the corpus and the alignment links.

3.2 Indexing

As mentioned previously, Cunei employs the same indexing approach used in Panlite, as this scales well with large amounts of data. The technique used in Panlite is to build a suffix array with the Burrows-Wheeler transform (Brown, 2004). Suffix arrays are an increasingly popular way to index large amounts of data and have been used as well by PSMT in (Zhang and Vogel, 2005) and (Callison-Burch, 2005). The Burrows-Wheeler transform brings the added benefit of considerably shrinking the size of the index.

In contrast to Panlite, Cunei needs to index the structure of the sentence as well as the lexical tokens. This was accomplished by using two indexes running in parallel as shown in Figure 3. Although this is not the most elegant approach, it is certainly the most practical approach. The two indexes allow for fast lookups of structural or lexical tokens. The downside is that the index is not optimized to look up combinations of structural and lexical tokens. To find the structural matches corresponding to a lexical match (or vice-versa), the sentence number and position within that sentence are identified and looked up in the other index.

For lookups in the index, the Burrows-Wheeler transform does not result in any increase in computation. However, if one desires to reconstruct the text from the index, then looking up each type requires an additional binary search. For this reason, Cunei stores the index as a Burrows-Wheeler transformed suffix array on disk, but also allows for run-time reconstruction of the original suffix array. To reconstruct the original suffix array is very fast (linear transformation) but does require more memory. This is only performed when the task at hand requires reconstructing large amounts of the text and continuously looking up each type creates a performance bottleneck. For translation, it is usually necessary to reconstruct the suffix array for the target side of the index, but not the source side of the index.

Another optimization made in Cunei is to represent the index as a memory-mapped bit array. The bit array is dynamically adjusted to use the minimum number of bytes that are capable of representing the total number of types and tokens present in the corpus. This allows for a much smaller data structure than just representing everything with an integer, and (in theory) has no upper bound. Furthermore, the memory-mapped nature of the file makes the load time significantly faster. In this work I indexed 100,000 sentence pairs which only took a few minutes and consumed 27.5MB in all (including lexical and structural types and tokens for source and target).

3.3 Alignment

The second major component of the system is alignment. GIZA++ (Och and Ney, 2003) was used to generate a word alignment over the entire cor-

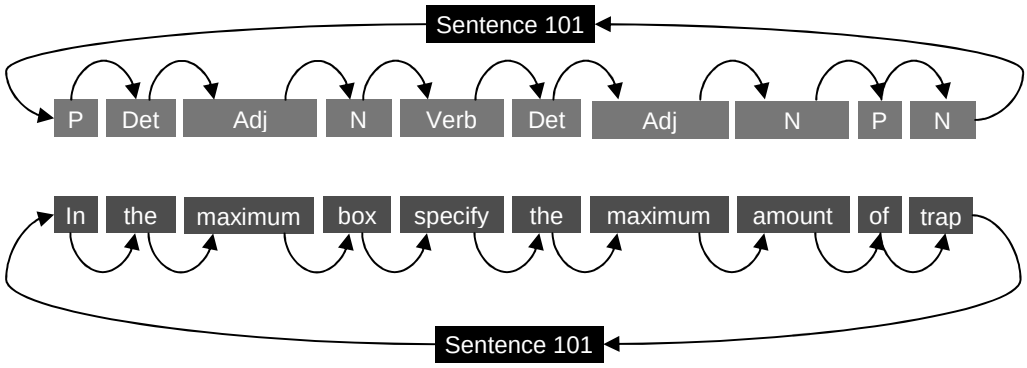


Figure 3. Indexing Structural and Lexical Tokens

pus. However, GIZA++ does not provide phrasal alignments which are necessary for translation. Thus, I investigated other alignment approaches and implemented a technique very similar to PESA (Vogel, 2005). The final alignment probability is calculated by taking the log-linear combination of the conditional probability of the entire source sentence given the target sentence, the conditional probability of the entire target sentence given the source sentence, and the length ratio between the selected source and target phrases. The conditional sentence probabilities are calculated by multiplying all the conditional word probabilities that agree with the phrasal alignment. A word alignment agrees with the phrasal alignment when it links two words that are both outside the phrasal alignment or two words that are inside the phrasal alignment.

3.4 Building Translations

Lexical translations are built by retrieving examples from the corpus and finding the aligned target text. Given a source text to translate, first Cunei looks in the source index for lexical examples of each sub-part of the source text. To ensure both speed and accuracy, a desired maximum number of instances of each distinct source phrase (typically 500-1000) is specified in a configuration file. If more than the desired number of examples are found, then the results are sub-sampled to only return the maximum. Each example is phrase aligned and the corresponding target text for each example is placed in a lattice. When more than one example produces the same target text, the results are merged together and their scores are combined. This is the same basic approach used in Panlite and PSMT systems with online alignment such as those described in (Zhang and Vogel, 2005) and (Callison-Burch, 2005).

Where Cunei differs from other systems is that after all lexical look ups have been performed, Cunei looks for structural matches. Recall that the preprocessing routine has already tagged the source text with part-of-speech tags. Cunei queries the structural source index for all part-of-speech sequences that match a section of the input text's structure. A structural example is skipped if it is less than three tokens long or the maximum number of lexical examples has already been found for that section. In either of these cases, there is reason to believe that structural matches will not be useful. Similar to the lexical translations, once an example is found, it needs to be aligned to the target text. In this case the alignment extracts the target part-of-speech sequence rather than the lexical tokens. The retrieved part-of-speech sequence is used to predict the structure of the lexical target. This target part-of-speech sequence is converted to lexical example(s) through substitution. By following the alignment links, lexical translations present in the lattice are substituted into the structural template to form a new lexical translation. All elements in the lattice are searched to build lexical translations such that they maintain the same structure and alignment links as found in the structural example. An example of this is demonstrated in Figure 4. While single word substitutions are the most common, this process also looks for entire phrases that form an appropriate substitution. Furthermore, structural matches are analyzed from

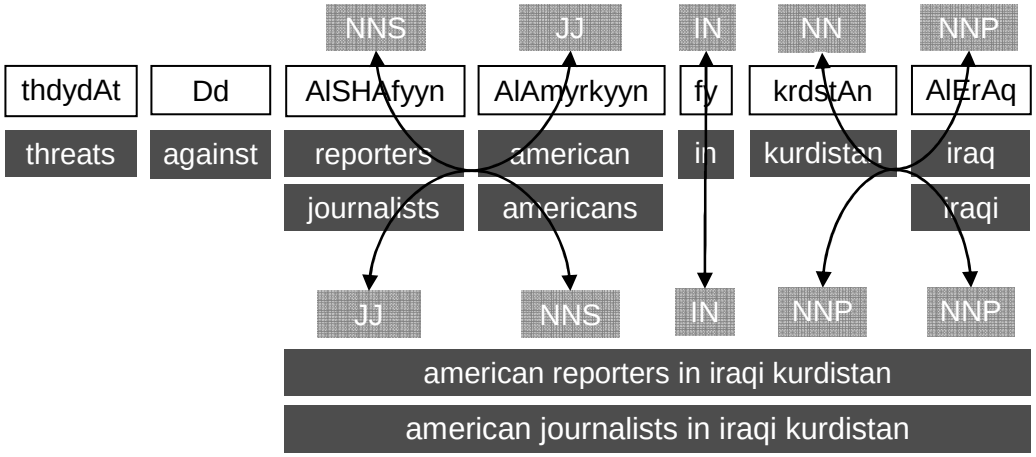


Figure 4. Example Constructed from Structural Template and Translation Lattice

shortest to longest so that longer matches can make use of translations created by shorter structural matches.

A minor exception to the process occurs when a structural example contains one or more lexical matches. To check for this situation, when a structural example is found, the lexical tokens of the structural example must be compared to the input text. When some of the lexical source tokens are the same, all target positions that align to a lexical source token are marked. These specially marked target positions cannot be replaced by other elements in the lattice. Rather, the lexical target tokens for these positions are retrieved from the corpus and used in the translation. This allows for structural examples where one or more source and target words are lexicalized even though the index does not directly support searching for this possibility.

3.5 Scoring Translations

Once all of the translations have been retrieved from the corpus or synthetically created from structural examples, it is necessary to score them. The language modeler will make the final decision as to which translations to use, but the language modeler must be provided with a score reflective of how likely each translation is to be representative of the source span it covers. In Cunei, each example that is placed in the lattice keeps track of three sub-scores: alignment probability, relative frequency, and context matches (the number of other examples in the lattice from the same sentence)[2]. When two translations are merged because they share the same target translation, their subscores are added together. A final score is produced by a log-linear combination of the three subscores which are averaged over all found translations. The weights of the log-linear combination are defined in a configuration file and are tuned using held-out data.

The synthetic lexical examples built by combining long structural examples and shorter lexical examples pose a problem for scoring. As this specific lexical translation never occurs in the corpus, it is difficult to determine its relative frequency—a critical component of the scoring. Furthermore, the distribution of the structural examples and the lexical examples is not the same, making the two relative frequencies hard to combine. Lastly, not all the structural examples are relevant to a particular input. Some structural examples can only occur with specific lexical elements or under specific conditions. Sometimes structural examples are found that cannot produce a lexical translation because the lattice lacks the necessary lexical items that match its structure and alignment constraints. Calculating relative frequency based on all the retrieved structural examples results in very low scores for each example, and it did not seem reasonable as many of these examples cannot occur for the given input.

To account for these differences, the relative frequencies for lexical and structural examples are calculated by only totaling over examples that produced a lexical translation. If the alignment process fails or if a structural example cannot find any appropriate lexical entries to create a lexical translation, then it is not included in the total count. In addition, a confidence score is applied to all translation candidates. If the translation candidate is retrieved from the corpus, then its confidence is 1.0. If the translation candidate is formed by a structural example, then its confidence score is the geometric mean of the scores of each lexical translation that was used (through substitution) to create the translation. This confidence score is an approximate measure of how closely a structural example matches the original source text. The confidence score is applied as a weight to each score when two translations are merged. Thus, an example with a low confidence score will not affect the overall scores as much as an example with a high confidence score. In practice this means that if a structural example predicts one target and a lexical example predicts a different target, the lexical example's target will have a higher score.

4 Results

Cunei was trained on approximately 100,000 sentence pairs (4.87 million words) of Arabic-English newswire text. This represents all available Arabic-English newswire text from the Linguistic Data Consortium with sentences containing fewer than 50 words. While more parallel Arabic-English data is available, most of it is out of domain and in the form of United Nations proceedings. The training

[2] Fully implemented in the system, but due to a significant slowdown in speed and very minor improvement in translation quality, the context score was disabled for the final results.

	MT03 (Tune)		News A		News B		Editorial		Speech		Full MT04	
Lexical and Reorder	0.444		0.483		0.455		0.321		0.339		0.397	
Structural and Reorder	0.452	1.65%	0.490	1.52%	0.475	4.38%	0.329	2.58%	0.364	7.52%	0.412	3.75%
Lexical no Reorder	0.419	-5.80%	0.461	-4.45%	0.434	-4.64%	0.320	-0.31%	0.333	-1.59%	0.385	-3.01%
Structural no Reorder	0.446	0.44%	0.490	1.51%	0.470	3.18%	0.333	3.83%	0.363	7.03%	0.411	3.57%

Figure 5. Table of Evaluation Results

data has good lexical coverage and at the same time is not prohibitively large for the structural matching.

Parameters for Cunei and the language modeler were tuned using part of the 2003 NIST MT Evaluation data set (MT03). However, due to time restraints, parameters for Cunei (as opposed to the language modeler) were not separately tuned for the system with structural matching enabled. Rather, I used the same parameters that were tuned on the system with structural matching disabled. Thus, these results do not reflect the full potential of the system with structural matching enabled.

Evaluation was performed by comparing Cunei with structural matching disabled to Cunei with structural matching enabled. This experiment was run twice: first with language model reordering enabled, and second with it disabled (monotonic decoding). All systems were evaluated on the 2004 NIST MT Evaluation data set (MT04), which provides five reference translations. MT04 contains editorial, speech, and news genres, but nearly half of it is news. I split MT04 by genre but also divided the news genre into two parts—one from Xinhua News Agency and the other from Agence France Press. Document boundaries were preserved in all the splits and the chunks range in size from 278 sentences to 387 sentences. Splitting the data in this fashion allowed multiple evaluations on different types of data while maintaining enough sentences to have meaningful results. In addition, a final score for all of MT04 is provided.

The results are shown in Figure 5 and Figure 6. It is clear that the structural matching improves translation quality as BLEU scores improved under all testing conditions. While the relative improvement is smallest for "News A", this is still a respectable gain in performance considering the high baseline. "News B", "Editorial", and "Speech", which all have lower baselines, show stronger gains from the structural matching. This correlates well to the initial hypothesis that structural matching will make the system more robust and allow it to degrade more gracefully.

As expected, when language model reordering is disabled, the performance of the system with only lexical matching drops. This is not true for the system with structural matching enabled—signifying that the structural matching is capturing most (if not all) of the reordering.

Figure 7 and Figure 8 illustrate visually the differences in the types of translations found between the lexical only system and the structural system.

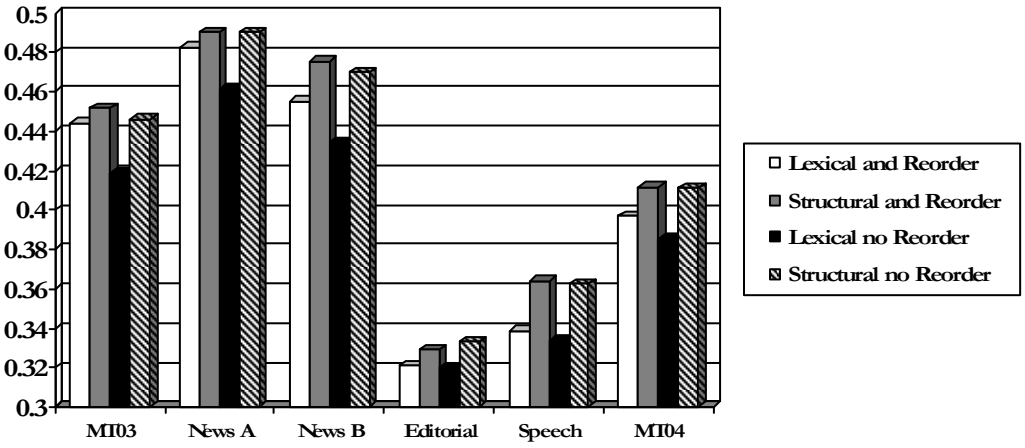


Figure 6. Chart of Evaluation Results

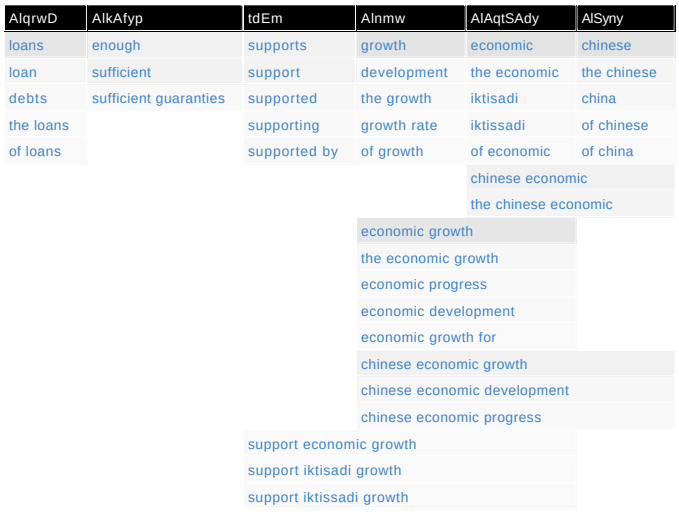


Figure 7. Translation Lattice with Structural Matching

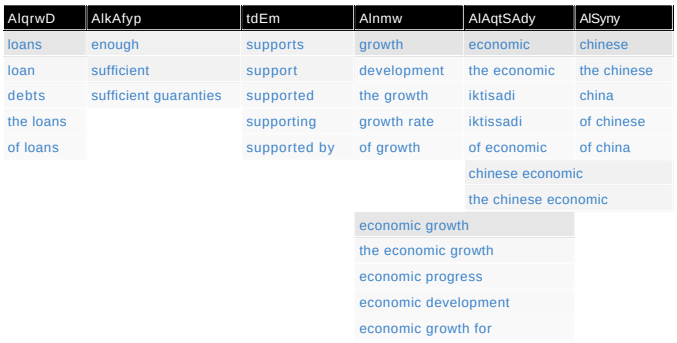


Figure 8. Translation Lattice without Structural Matching

5 Remaining Issues and Future Work

The problem of combining scores from two different probability distributions is fundamentally hard and the solution is not readily apparent. Applying confidence weights seemed reasonable, but I imagine much better solutions exist. Even if the confidence weights were retained, it would be worthwhile to investigate applying them in a non-linear fashion. Time limitations prevented experimentation with other methods.

Figure 7 and Figure 8 illustrate another problem: phrases inserted into the lattice do not always have optimal boundaries. The last three words "Alnmw AlAqtSAdy AlSyny" form one noun phrase that translates as "chinese economic growth". The lexical system only provides "economic growth" and "chinese economic". The structural matching does create "chinese economic growth", but it also has partial translations of "economic growth", "chinese economic", and "support economic growth". The problem is that these partial translations sometimes inappropriately guide the language modeler. Both the lexical and structural systems are affected by this issue, but the problem occurs with greater frequency when structural matches are enabled. This problem brings up the question of what makes a suitable translation unit. I did experiment with restrictions similar to those in the Gaijin system by limiting which part-of-speech tags a phrase is allowed to begin and end with. However, all of these experiments that "filtered" the lattice resulted in lower scores. It would be worthwhile to investigate how to select more appropriate translation units, but in the meantime it appears to do more good than harm to allow all possible phrases.

Perhaps the most apparent "problem" with forming lexical translations from structural examples is speed. Enabling structural matching significantly slows down the system. It is for this reason that I did not tune all the parameters of the structural engine. The problem is that there are usually a lot of structural examples found in the corpus, and there are also a multitude of lexical translations that can be substituted into each structural example. The issue with speed is not due to poorly written code, but to the thousands of combinations that need to be analyzed for a match per example. The longer the example is, the more prone it is to this problem. I have partially alleviated this problem by pruning and chunking the input into smaller units. However, this merely makes the computation tractable, and not fast. More aggressive pruning and/or heavy caching techniques truly should be investigated.

6 Conclusion

In conclusion, this research describes a system that synthesizes two different approaches to EBMT. Whereas the origins of this system lie with EBMT, the end result is hard to classify as an EBMT system. Cunei has borrowed heavily from ideas and techniques present in EBMT, PSMT, and Xfer.

What is clear from this work, however, is that a data-driven approach that combines exact lexical matching with structural templates improves translation quality.

Acknowledgements

I would like to thank Ralf Brown for the reviewing this paper and his insightful thoughts throughout the research.

References

Ralf D. Brown. 1996. Example-Based Machine Translation in the Pangloss System. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, pp. 169--174,

Ralf D. Brown. 2004. A Modified Burrows-Wheeler Transform for Highly-Scalable Example-Based Translation. In *Machine Translation: From Real Users to Research, Proceedings of the 6th Conference of the Association for Machine Translation*, Washington, DC, pp. 27--36.

Chris Callison-Burch, Colin Bannard and Josh Schroeder. 2005. Scaling Phrase-Based Statistical Machine Translation to Larger Corpora and Longer Phrases. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, Michigan, pp. 255--262.

Jaime Carbonell, Katharina Probst, Erik Peterson, Christian Monson, Alon Lavie, Ralf Brown, Lori Levin. 2002. Automatic Rule Learning for Resource Limited MT. In *Proceedings of 5th Conference of the Association for Machine Translation in the Americas*, Tiburon, California, pp. 1--10.

Mona Diab, Kadri Hacioglu and Daniel Jurafsky. 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In *Proceedings of HLT-NAACL 2004*.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1): pp. 19--51.

Katharina Probst and Lori Levin and Erik Peterson and Alon Lavie and Jaime Carbonell. 2003. MT for Minority Languages Using Elicitation-Based Learning of Syntactic Transfer Rules. *Machine Translation*, 17(4): pp. 245--270.

Adwait Ratnaparkhi. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 133--142.

Tony Veale and Andy Way. 1997. Gaijin: A Template-Driven Bootstrapping Approach to Example-Based Machine Translation. In *Proceedings of NeMNLP97, New Methods in Natural Language Processing*, Sofia, Bulgaria.

Stephan Vogel. 2005. PESA: Phrase Pair Extraction as Sentence Splitting. In *Proceedings: The Tenth Machine Translation Summit*, Phuket, Thailand.

Ying Zhang and Stephan Vogel. 2005. An Efficient Phrase-to-Phrase Alignment Model for Arbitrarily Long Phrase and Large Corpora. In *Proceedings of the Tenth Conference of the European Association for Machine Translation*, Budapest, Hungary.

Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model

Kay Rottmann
InterACT
Interactive System Labs
University of Karlsruhe
Am Fasanengarten 5
76131 Karlsruhe, Germany
rottmann@ira.uka.de

Stephan Vogel
InterACT
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Av.
Pittsburgh, PA 15213
vogel+@cs.cmu.edu

Abstract

In this paper we describe a word reordering strategy for statistical machine translation that reorders the source side based on Part of Speech (POS) information. Reordering rules are learned from the word aligned corpus. Reordering is integrated into the decoding process by constructing a lattice, which contains all word reorderings according to the reordering rules. Probabilities are assigned to the different reorderings. On this lattice monotone decoding is performed. This reordering strategy is compared with our previous reordering strategy, which looks at all permutations within a sliding window. We extend reordering rules by adding context information. Phrase translation pairs are learned from the original corpus and from a reordered source corpus to better capture the reordered word sequences at decoding time. Results are presented for English → Spanish and German ↔ English translations, using the European Parliament Plenary Sessions corpus.

1 Introduction

Statistical machine translation (SMT) is currently the most promising approach to large vocabulary text translation. In the spirit of the Candide system developed in the early 90s at IBM (Brown et al., 1993), a number of statistical machine translation systems have been presented in the last few years (Wang and Waibel, 98), (Och and Ney., 2000), (Yamada and Knight, 2000), (Vogel et al., 2003). These systems share the basic underlying principles of applying a translation model to capture the lexical and word reordering relationships between two languages, complemented by a target language model to drive the search process through translation model hypotheses. The reordering of words in machine translation still remains one of the hardest problems. Here we will describe our approach using syntax-based reordering rules to create a lattice structure for test sentences that encodes all word reorderings consistent with the reordering rules learned from a word aligned training corpus.

2 Modeling Word Reordering

Different languages differ in their syntactic structure. These differences in word order can be local or global. Local reorderings are for example the swapping of adjective and noun in language pairs like Spanish and English:

Example: ADJ NN → NN ADJ
An important agreement
Un acuerto importante

Word order changes which span across the entire sentence pose a much tougher problem. For example, in the translation from German to English especially verbs participate in long range reorderings.

Example: auxiliary verb and infinite verb
Ich *werde* morgen nachmittag ... *ankommen*
I *will arrive* tomorrow afternoon ...

The '...' indicates that other information (eg. 'mit dem Zug' → 'by train') could be embedded, pushing the auxiliary verb and the infinite verb even apart.

Another example of long-distance reordering is the detached verb prefix in German.

Example: detached verb prefix
Ich *komme* morgen nachmittag ... *an*.
I will *arrive* tommorow afternoon ...

The verb prefix 'an' is detached from the main verb 'komme' and moved to the end of the sentence. It is difficult to generate 'arrive' from 'komme' in a phrase-based system. Even more difficult is the translation from English into German, where arrive needs to generate both 'arrive' and 'an' at different positions in the target sentence.

To generate the correct word sequence the translation system needs to have strong, restricting evidence of how to rearrange the words, this is the approach taken in grammar-based systems, or it has to have weak evidence in the form of probabilities, and then test all (or at least a large number) of reorderings, as is the strategy in typical phrase-based statistical translation systems.

The well-known IBM and HMM word alignment models (Brown et al., 1993) and (Vogel et al., 1996) contain as one component a so-called distortion model to capture the different word orders in different languages. These distortion models can be formulated in terms of absolute positions, as in the IBM2 model, or in terms of relative positions, as in the HMM and IBM4 alignment models. These distortion models are rather weak. They essentially boil down to saying that long distance reorderings are less likely then short distance reorderings.

It is important to notice that these distortion models do not pose any restrictions as to which reorderings are possible. At decoding time all permutations need to be considered, which is impossible for any but very short sentences. A restriction to word reordering was introduced in (Wu, 95). The ITG (inverse transduction grammar) constraint allows only reorderings, which can be generated by swapping subtrees in a binary branching tree. Still, for longer sentences the number of possible reorderings is too large to be enumerated; severe pruning is necessary.

To make the distortion models more informative the aligned positions can be conditioned on the length of the sentences, on the words (lexicalized distortion models), or on word classes (parts-of-speech) or automatically generated word classes, using clustering techniques (Al-Onaizan and Papineno, 2006).

State-of-the-art SMT systems use phrases. One advantage is that phrases can capture some of the local reordering patterns. However, this is rather limited as the average length of matching phrases is typically less then two words. To capture longer ranging word reorderings these phrases need to be reordered, which brings us back to the central questions:

- How to model word reordering?
- How to estimate the parameters of the model?
- How to apply the model at translation (decoding) time?

These questions will –at least to some extent– be dealt with in subsequent sections.

2.1 Related Work

Different approaches have been developed to deal with the word order problem. First approaches worked by constraining reorderings at decoding time (Berger et al., 1996). In (Wu, 1996) the alignment model already introduces restrictions in word order, which leads also to restrictions at decoding time. A comparison of these two approaches can be found in (Zens and Ney, 2003). They have in common that they do not use any syntactic or lexical information, therefore they rely on a strong language model or on long phrases to get the right word order. Other approaches were introduced that use more linguistic knowledge, for example the use of bitext grammars that allow parsing the source and target language (Wu, 1997). In (Shen et al., 2004) and (Och et al., 2004) syntactic information was used to rerank the output of a translation system with the idea of accounting for different reordering at this stage. In (Tillmann and Zhang, 2005) and (Koehn et al., 2005) a lexicalised block-oriented reordering model is proposed that decides for a given

phrase whether the next phrase should be oriented to its left or right.

The most recent and very promising approaches that have been demonstrated, reorder the source sentences based on rules learned from an aligned training corpus with a POS-tagged source side (Chen et al., 2006), (Popovic and Ney, 2006) and (Crego and Marino, 2006). These rules are then used to reorder the word sequence in the most likely way.

3 Syntactic Reordering Rules

In our approach we follow the idea proposed in (Crego and Marino, 2006) of using a parallel training corpus with a tagged source side to extract rules which allow a reordering before the translation task. By doing it this way we are able to keep the translation process in the decoder monotone and make it significantly faster compared to allowing reorderings in the decoder. To avoid making any hard decisions in reordering the source side we use a lattice structure as input (Crego and Marino, 2006), (Zhang et al., 2007) for our decoder. Lattices are created for the source sentences and contain all the possible reorderings and of course also the original word sequence. As a new feature we use the context in which a reordering pattern is seen in the training data. Context refers to the words or tags to the left or to the right of the sequence for which a reordering has been observed. By doing this we hope to differentiate between reorderings that are dependent on their context.

3.1 Learning Reordering Rules

The rules that are later applied to the source sentences are learned via an aligned corpus for which the POS information of the source sentences is available. Given a sentence pair with source words f_1^J and target words e_1^I, and the alignment a_1^J a reordering rule is extracted whenever the alignment contains a crossing, i.e. whenever there is i and j with $i < j$ and $a_i > a_j$. Within one sentence pair we always extract the longest reordering sequences only. A rule, which is observed as part of a longer reordering, is only stored if it also occurs as the longest reordering sequence in some other sentence pair. The motivation for this is that only those reorderings get learned, which really exist for themselves. This restriction allows us to extract longer reordering patterns and still keeping the number of reordering patterns manageable. This will also restrict the application of rules in wrong place in the later reordering approach.

In a second step of learning, relative frequencies are computed for every rule that has been observed more than a given number of times in the training corpus (we observed good results with more than 5 times). Because the number of rules is very high, a Suffix-Array (Zhang and Vogel, 2006) is used for faster computation of the occurrence-counts for the observed sequences that triggered a reordering.

By the above described mechanisms, we are able to extract rules using as a trigger for the reordering of the words the following types.

- Tag sequence
- Word sequence
- Context of one or two tags before and / or after the Tag sequence
- One or two words before and / or after the Tag sequence

Table 1 shows examples for rules consisting of the plain tag sequence and rules that use an additional (left) context separated by the '::'. The final reordering rule consists of the source side sequence of POS tags or words that trigger a reordering, the permutation of this sequence (given as the numbers indicating the reordering) and the relative frequency of this reordering given the source sequence in the training corpus.

source sequence	rule	freq.
PDAT NN VVINF	3 1 2	0.60
VAFIN :: PDAT NN VVINF	3 1 2	0.63
KOUI :: PDAT NN VVINF	3 2 2	0.88
moechte :: PDAT NN VVINF	3 1 2	0.92

Table 1: Example rules for German to English translation with no context, with one tag of context to the left and one word of context to the left

All four rules in Table 1 reorder the same sequence (moving the infinite Verb to the front),

with different relative frequencies assigned to them. The first entry uses no context information, while the other 3 lines show the rules with context information – in this case a left context only. For this POS pattern the strongest evidence for a reordering comes from the tag sequence with one source word in front of the reordering.

3.2 Applying Reordering Rules

We begin with a lattice that contains only the monotone path of the sentence that has to be translated. First, the POS tagging is done. Then, for every sequence of POS up to a maximum length (20 in our experiments) it is tested if it occurs as the left-hand side of any reordering rule. If a match is found, then for each right-hand side a new path is added to the lattice with the words now in the reordered sequence. Similarly, for POS sequences plus left/right context, which can be POS tags or words, if a match is found then a new path is added to the lattice. This also covers the reordered part only and ignores the context positions.

To guide the decoder through the lattice by favoring often seen reorderings the relative frequency of every reordering rule is applied to the first edge after a node where the path splits up. In this case it is important to know how the scores are applied to the edges. Since we used different type of rules the relative frequencies do not sum up to 1 over all rules, but only over the rules of one type.

Another problem is introduced by the fact that the reorderings are of different lengths, and only reorderings over the same length are comparable in their scores.

So we decided to score at the outgoing edges of a node, first scoring the longer reorderings and then using the remaining probability mass for the shorter reorderings. That means for one type of rule the score of a reordering in the lattice is its relative frequency seen in the training corpus weighted with the remaining probability mass of the monotone subpath where it takes place. In detail, for reordering subpath p via the m'th of n applied rules from node l to node r for this subpath, the scores are modified and the sum over all scores of edges going out of a node sums up to 1. In the following $P(p_m)$ denotes the relative frequency for the reordering p_m.

$$Score(p_m^{l,r}) = ProbabilityMass^{l,r} \cdot P(p_m)$$

where $ProbabilityMass^{l,r}$ is the probability mass that is remaining for the monotone subsequence from node l to node r. The effective score for the monotone path then computes

$$Score(monotone^{l,r}) =$$

$$ProbabilityMass^{l,r} - \sum_{i=1}^{n} Score(p_i^{l,r})$$

so that the $ProbabilityMass$ left on the subpath from l to $r-1$ is the $Score(monotone^{l,r})$. Figure 1 shows a small example lattice with only one applied rule, and Figure 2 a lattice with more applied rules.

The next step is to combine the scores of rules with different types of context. Those rules all have different relative frequencies, that are not comparable. A high relative frequency however means that this kind of reordering was seen very often during training. So we decided to compute the scores for the rules of different context by their own, only using rules of the same context. Then we applied to a reordering that was seen by more than one ruletype, that score which was the maximum for that rule. This ensures, that those reorderings that are triggered because they occur in a special context are favored. The monotone path however, gets the minimum of all scores computed for the monotone path over the different context rules.

4 Experiments

To study the effect of the POS-based distortion model we did a number of experiments on German-to-English, English-to-German, and English-to-Spanish translation tasks. We used the European Parliament Speeches Corpus as used in the TC-Star[1] project and the SMT-Workshop evaluations. Some details of the corpus are given in Table 2.

Here train-xx is the complete training corpus, dev-xx denotes the development test set used for the MER-training (Och, 2003), and eval-xx is the unseen test set used for evaluation. In the case of

[1] http://www.tc-star.org

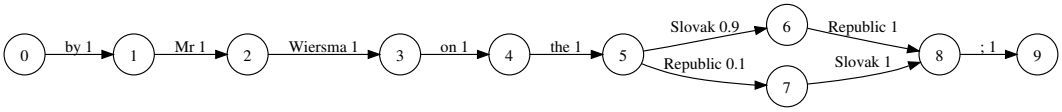


Figure 1: Example for a very small reordering lattice

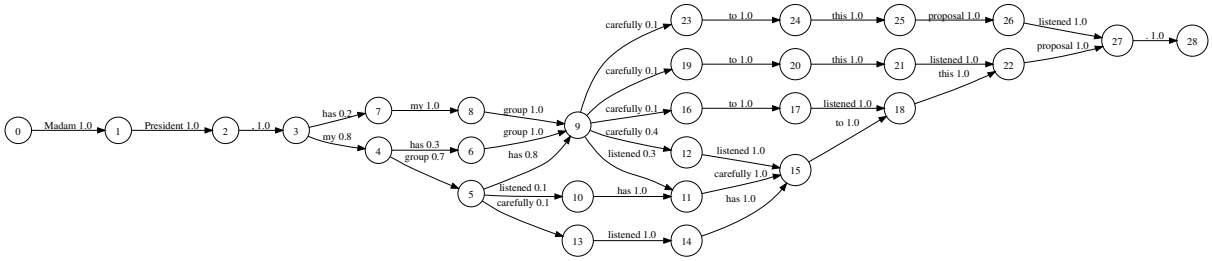


Figure 2: a larger lattice example

	Sentences	Words	Voc/OOV
train-en	1.2M	35M	97K
train-de	1.2M	33M	298K
dev-en	2K	58K	6103 / 62
dev-de	2K	54K	8762 / 306
eval-en	2K	58K	6246 / 250
eval-de	2K	55K	9008 / 551
train-en	1.2M	33M	94K
train-es	1.2M	34M	135K
dev-en	1.2K	30K	4084 / 79
eval-en	1.1K	30K	4100 / 105

Table 2: Corpus statistics EPPS training and test corpora.

German $\leftrightarrow$ English translation the evaluation is based on 1 reference, for English $\rightarrow$ Spanish on 2 references.

For the alignment and the phrase extraction we used the Pharaoh training package (Koehn et al., 2005). To tag the corpora we used the following taggers: for English the Brill tagger (Brill, 1995) with a tag set size of 36 and for German the Stuttgart tree-tagger with a tag set size of 57 tags (Schmid, 1994). From the training corpora and the POS tagged source side we extracted the reordering rules according to the method described in Section 3.1. For the experiments reported in this paper we only learned rules up to a length of 15, since longer rules do not occur often enough in the training corpus. Table 3 displays the counts of rules that consist only of the tag sequence and those that use additional context with the tag to the left and the tag to the right learned from the training data as well as the number of rule usage on the test sentences.

4.1 Threshold and Context

In the first series of experiments we wanted to study two questions: how does the threshold value for the relative frequencies of the rules affect the translation quality, and is using context for the reordering patterns helpful. For the influence of the context we used only those rules that used the tags to the left and to the right of a reordered tag sequence. We chose that kind of context for this task because although it would probably perform worse than no context, it would indicate, which threshold is best for both types of context, those only before the reordering sequence and those after the sequence. Higher threshold, i.e. fewer rules should eventually hurt the performance. On the other side, allowing unreliable reordering rules to be used could also lead to a degradation. The results for those experiments can be seen in Table 4 and in Table 5.

The systems named *POS no Context* are those that only use the tag sequence for triggering reorderings, while those named *POS + Context* use only rules with left and right tags as context. The value behind the system name indicates the relative frequency threshold for the rules. All BLEU scores are for case sensitive evaluation. As a base-

System		# en → es		# en → de		# de → en	
Context	Threshold	Rules Learned	Rule Matches	Rules Learned	Rule Matches	Rules Learned	Rule Matches
no	0.05	21388	12715	7929	60692	13396	72728
	0.1	6848	7740	4061	27809	8528	32233
	0.2	2321	4247	1291	8192	3738	14615
	0.3	1136	3369	469	3879	1601	7076
yes	0.01	72772	21119	32380	89225	38858	88549
	0.05	46014	6888	22836	36765	28485	37608
	0.1	25962	4924	15941	19319	21469	17148
	0.2	15304	3461	8462	8574	14466	9534

Table 3: Number of reordering rules learned from the training corpus and number of rule matches on the test sentences with respect to the relative frequency threshold, without and with using the context POS tags

System	en → es
Baseline(RO3)	49.98
POS no Context 0.05	50.36
POS no Context 0.1	51.09
POS no Context 0.2	50.66
POS no Context 0.3	50.59
POS + Context 0.01	50.92
POS + Context 0.05	50.90
POS + Context 0.1	50.84
POS + Context 0.2	50.74
unseen Baseline(RO3)	48.51
unseen no Context	49.57
unseen with Context	49.49

Table 4: Case sensitive BLEU scores on English to Spanish development and test sets for the different applied threshold values

System	en→ de	de → en
Baseline(RO3)	18.92	25.64
POS no Context 0.05	19.48	26.69
POS no Context 0.1	19.55	26.46
POS no Context 0.2	19.30	26.01
POS no Context 0.3	19.22	25.73
POS + Context 0.01	19.34	25.85
POS + Context 0.05	19.34	25.86
POS + Context 0.1	19.44	25.79
unseen Baseline(RO3)	17.69	23.70
unseen no Context	17.78	24.79
unseen with Context	17.79	23.87

Table 5: Case sensitive BLEU scores on English and German development sets for the different applied threshold values

line we used our decoder with internal reordering (Vogel, 2003). The internal reordering was deactivated for every other system. So the scores reported for the reordering using the POS information does not use any additional internal reordering.

Although the first series of experiments was conducted on the developement set, it is possible to draw some conclusions from the observed results. Somewhat surprising is the fact that the system that used only the rules with context for the English to Spanish task was nearly as good as the system that did not use any context. The results get even more surprising, if you review the number of rules that were used to generate the lattices (Table:3). With a threshold value of 0.05 the number of rules with context that were applied, were even lower than the number of rules for the best setting without context while achieving nearly the same BLEU score. This means that the rules with context are able to cover as many reorderings as the rules without context although they are more specific. From this it can be seen that the reorderings in the translation from English to Spanish often occur in the same context.

In the English and German translations however, the situation is quite different. Here the

score with the rules that make use of context information is below the scores without context information by ≈ 0.2 BLEU points. This is what we expected, since the German language allows a lot of reorderings of the same word sequence, because this type of context of reorderings in the German language varies a lot and it is hard to extract specific rules without omitting others. However the number of rules for the best settings with and without context shows that the system without context applied 50% more rules to the devset, which also shows the more general form of the rules without context.

Nevertheless there are some reorderings in the German language that suggest that some rules require context information. For example in sentences with auxiliary verbs, it is possible to learn a rule that moves the verb to the auxiliary verb which stays in place (e.g. " Er hat ... gesagt."). Without context it is not possible to cover those dependencies without a huge increase of wrong reorderings or the score for such a reordering is much to low to get ever applied.

Using the best system tunded on the developement data for the unseen data provided a nice improvement over the baseline system and even the system that used the context of the left and right tags performed in all three tests on the unseen data better than the internal reordering. This along with the results we observed indicate that while some reordering are better covered when context information is used, there are some reordering for which no context is useful.

In order to utilize this, we built reordering lattices that contained reorderings triggered by all extracted rules, not only just one type (Table 6 and Table 7). One problem which arose was that the rules that only used the source word sequence and no POS information hurt performance. This is obvious, since these rules only get learned if the word sequence appears often enough in the training corpus. The problem is that this however also leads to good phrases for these sequences. By having high probability reorderings for those sequences, those phrases that provide the good translation are not useful anymore and the performance is hurt.

Overall the results show that the approach of using syntactic reordering outperforms the internal reordering. In all tested language pairs we saw an improvement: in the German do English and the English to Spanish task the improvement was more than 1.0 BLEU. Also the combination of rules with different context types can lead to better performance. The improvement achieved over a single type of rule depends on the language pair, but for the translation task from English to Spanish we saw an improvement of more than 0.3 BLEU and for English to German it was more than 0.4 BLEU. In the German to English task the Improvement was only 0.1 BLEU.

System	en → es
unseen Baseline(RO3)	48.51
unseen no Context	49.52
unseen with Context	49.49
unseen combination	49.58
unseen combination-Lex	49.83

Table 6: Case sensitive BLEU scores on English to Spanish translation with with combination of all rule types and all rules except those that use only source words as trigger

System	en→ de	de → en
unseen Baseline(RO3)	17.69	23.70
unseen no Context	17.78	24.79
unseen with Context	17.79	23.87
unseen combination	18.27	24.85
unseen combination-Lex	18.21	24.88

Table 7: Case sensitive BLEU scores on English and German translation with combination of all rule types and all rules except those that use only source words as trigger

4.2 Reordering the Training Corpus

The next series of experiments we tried examined the influence of reordering in the training corpus (Popovic and Ney, 2006). One main reason why this should lead to further improvement lies in the the observation we made above, that often seen rules may contradict phrases. This effect can be seen most significantly when looking at the performance with and without rules that are only based on the exact word sequence on

Corpus	en → de	de → en
Combination	19.61	26.88
Reordered (Giza)	19.44	26.76
Reordered (Lattice)	20.00	27.06
unseen Baseline(RO3)	17.69	23.70
unseen combination	18.27	24.85
unseen reordered corpus	18.42	25.06

Table 8: Case sensitive BLEU scores using phrases from reordered training corpus

the source side. (Popovic and Ney, 2006) also reported improvements when reordering the training corpus. We conducted experiments on the English to German and German to English translation task and tried two different ways of reordering the training corpus.

The first way was to extract phrases from a corpus that had been reordered based on the existing alignment information. That is to say, the source sentence was reordered to make the alignment between source and target sentence monotone.

The second approach we tested was using the learned reordering rules to create a reordering lattice for every source sentence. Then we used the word sequence on the best path, i.e. the path with the highest score, as new source sentence. The scores we used for the edges were the same as described above. After reordering the source corpus we used this to extract a new phrase table. The results of the tests can be seen in Table 8.

As it can be seen in Table 8, the phrases extracted from the reordered training corpus using the alignment information directly performed worse than those phrases that were obtained from the corpus that was reordered using the reordering lattices.

On the unseen test data, we see an improvement of 0.15 in BLEU score compared to the previously best configuration for English to German and an improvement of 0.2 for German to English. So we were able to reproduce the effect reported by (Popovic and Ney, 2006), that a reordered training Corpus leads to a further improvement of the translation quality. As a result you can say that using the same reordering strategy for the training data as for the test data is preferable over just reordering the training corpus based on the word alignment generated by the word alignment models.

5 Future work

In the future we will try to minimize the rules that are applied to a test set for further reduction of the runtime. We believe the way to achieve this is by a better estimation of the scores for the monotone path and by alternative scoring methods so that effective pruning can be done. Also the effect of smoothing the relative frequencies should be revisited for the reordering rules.

One question that has not been answered yet, is whether additional decoder-internal reordering is still helpful. Some experiments have indicated this, and the effect seems to depend on the language pair. Another field we are working on is the integration of long range reordering rules (e.g. of the form: AUX * VB - 0 2 1, which would allow in German to English translations to move a verb next to the corresponding auxiliary verb). This can be done via the above stated rules, or as a combination with chunk reordering (Zhang et al., 2007). In the experiments described in the paper we relied on existing POS taggers. An alternative would be to use automatic clustering to obtain word classes. This would especially be useful when dealing with languages for which no good POS taggers are available. First experiments on applying word clustering for that task seem to be promising.

6 Conclusions

We presented a reordering model based on rules learned from a tagged aligned corpus. The results we obtain show that this approach outperforms our previous word reordering strategy, which used only distance information. We presented results on English to Spanish translation, which showed improvements of up to 1.3 BLEU points on unseen test data. For German to English and English to German the improvements where 0.6 and 1.1 BLEU point respectively on unseen data.

Furthermore we investigated the effect of extracting the phrase table from an reordered training corpus. By doing so we were able to obtain an additional improvement on the tested language

pair German to English and English to German. So overall the improvement of the German to English translation added up to 0.8 BLEU points over the baseline result and the total improvement from English to German was 1.3 BLEU points.

It is important to note that there was no further internal reordering applied when translating the lattices - so this can possibly lead to a further performance boost. The translation time we observed was in all settings ≈ 2 times faster than the approach of reordering only in the decoder. This is due to the monotone decoding over the lattice. Some sample translations of the baseline system with internal reordering, the system with POS-reordering without context and the combination of POS-reordering with and without context can be seen in Table 9.

7 Acknowledgements

This work was partly funded by the National Science Foundation under the project STR-DUST (Grant IIS-0325905) and by DARPA under the GALE project.

References

Yaser Al-Onaizan and Kishore Papineno. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 4th annual meeting of the ACL*, pages 529–536, Sydney, Australia.

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39.

Eric Brill. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4):543–565.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263.

B. Chen, M. Cettolo, and M. Federico. 2006. Reordering rules for phrase-based statistical machine translation. In *Int. Workshop on Spoken Language Translation Evaluation Campaign on Spoken Language Translation*, pages 1–15.

Josep M. Crego and Jose B. Marino. 2006. Reordering Experiments for N-Gram-based SMT. In *Spoken Language Technology Workshop*, pages 242–245, Palm Beach, Aruba.

P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *ACL 2000*, pages 440–447.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. 2004. A smorgasbord of features for statistical machine translation. *Proc. 2004 HLT-NAACL*, page 161.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

M. Popovic and H. Ney. 2006. POS-based word reorderings for statistical machine translation. In *Proc. of the 5th Int. Conf. on Language Resources and Evaluation (LREC)*, page 1278, Genoa, Italy.

H. Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

L. Shen, A. Sarkar, and F. J. Och. 2004. Discriminative reranking for machine translation. *In HLT-NAACL 2004: Main Proc.*, page 177.

C. Tillmann and T. Zhang. 2005. A localized prediction model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 557–564, Ann Arbor, MI.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-Based Word Alignment in Statistical Translation. In *COLING 16*, pages 836–841.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, and Alex Waibel. 2003. The CMU statistical translation system. In *Proceedings of MT Summit IX*, New Orleans, LA, September.

Stephan Vogel. 2003. SMT decoder dissected: Word reordering. In *Proceedings of the Int. Conf. on Natural Language Processing and Knowledge Engineering (NLP-KE)*, pages 561–566, Beijing, China, October.

System	Translation
English Source	. . . - which we chose to set up -to continue to play a full role in this area .
Baseline	. . . , die wir haben eingerichtet, um weiterhin eine vollwertige Rolle spielen in diesem Bereich.
POS	. . . , die wir haben eingerichtet, um weiterhin eine umfassende Rolle in diesem Bereich spielen.
Combination	. . . , die wir festgelegt haben, weiterhin eine umfassende Rolle in diesem Bereich spielen .
German Source	. . . geschah, bevor das Umweltbewusstsein ausreichend geschaerft war und ehe man wusste , welche Auswirkungen das haben wuerde.
Baseline	. . . happened before the increased environmental awareness sufficient was and before we knew what impact this would have .
POS	. . . happened before the environmental awareness sufficient was and before we knew what the impact of the would have .
Combination	. . . happened before the environmental awareness was sufficient and before we knew what impact this would have .

Table 9: Sample translations of different system types

Yeyi Wang and Alex Waibel. 98. Fast Decoding for Statistical Machine Translation. In *Proc. ICSLP 98*, pages 2775–2778.

D. Wu. 1996. A polynomial-time algorithm for statistical machine translation. *Proc. 34th Annual Meeting of the Assoc. for Computational Linguistics*, page 152.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377.

Kenji Yamada and Kevin Knight. 2000. A Syntax-based Statistical Translation Model. *ACL 2000*.

R. Zens and H. Ney. 2003. A comparative study on reordering constraints in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1, pages 144–151, Sapporo, Japan.

Ying Zhang and Stephan Vogel. 2006. Suffix array and its applications in empirical natural language processing. Technical Report CMU-LTI-06-010, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Dec.

Y. Zhang, R. Zens, and H. Ney. 2007. Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Syntax and Structure in Statistical Translation (SSST)*, pages 1–8, Rochester, NY.

Automatic induction of shallow-transfer rules for open-source machine translation

Felipe Sánchez-Martínez and **Mikel L. Forcada**
Transducens Group Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant, Spain
{fsanchez,mlf}@dlsi.ua.es

Abstract

This paper focuses on the inference of structural transfer rules for shallow-transfer machine translation (MT). Transfer rules are generated from alignment templates, like those used in statistical MT, that have been extracted from parallel corpora and extended with a set of restrictions that control their application. The experiments conducted using the open-source MT platform Apertium show that there is a clear improvement in translation quality as compared to word-for-word translation (when no transfer rules are used), and that the resulting translation quality is very close to the one obtained using hand-coded transfer rules. The method we present is entirely unsupervised and benefits from information in the rest of modules of the MT system in which the inferred rules are applied.

1 Introduction

The increasing availability of machine-readable parallel corpora has given rise to the development of corpus-based machine translation (MT) approaches such as statistical MT (SMT) or example-based MT (EBMT). However, corpus-based approaches usually require a very large parallel corpus (with tens of millions of words) that is not always available. On the other hand, rule-based MT (RBMT) attains high performance but at the expense of the large effort needed to build the necessary linguistic resources (Arnold, 2003) such as structural transfer rules.

In this paper we focus on the automatic inference of structural transfer rules from parallel corpora, which are small compared to the size of corpora commonly used to build SMT or (some) EBMT systems. The approach we present is tested on the shallow transfer MT platform Apertium for which structural transfer rules are generated.

Overview. In rule-based MT, transfer rules are needed to perform syntactic and lexical changes. The approach we present in this paper to infer shallow-transfer MT rules is based on the alignment templates approach (Och and Ney, 2004) already used in SMT (see section 3). An alignment template (AT) can be defined as a generalization performed over aligned phrase[1] pairs (or *translation units*) by using word classes.

The method we present is entirely unsupervised and needs, in addition to the linguistic data used by the MT system in which the inferred rules are used, only a (comparatively) small parallel corpus and a file defining a reduced set of lexical categories usually involved in lexical changes.

Sanchez-Martınez and Ney (2006) use ATs to infer shallow-transfer rules to be used in

[1]For the purpose of this paper, with *phrase* we mean any sequence of consecutive words, not necessarily whole syntactic constituents.

MT. The work reported in this paper can be seen as a reformulation and improvement of that work. Sanchez-Martınez and Ney (2006) use ad-hoc linguistic information, in addition to that already present in the rest of modules of the MT system, to define the priorities used to establish agreement restrictions. This additional linguistic information is not necessary here, as restrictions may be easily derived from the bilingual dictionary using a general approach.

Transfer rules are generated for use with the open-source shallow-transfer MT platform Apertium; however, the approach we present is suitable for any other shallow-transfer-based MT system. The generated transfer rules (see section 2.1) are coded in a well-defined XML format, and can be edited by human experts or even co-exist with handcrafted ones.

The method we present[2] has been tested with an Apertium-based MT system for the Spanish–Catalan language pair; the experimental results show that the use of AT-based shallow-transfer rules drastically improves the translation quality as compared to word-for-word translation, i.e. when no transfer rules are used, and is comparable to the quality achieved when using handcrafted rules.

Background. There have been other attempts to learn automatically or semi-automatically the structural transformations needed to produce correct translations into the target language (TL). Those approaches can be classified according to the translation framework to which the learned rules are applied. On the one hand, some approaches learn transfer rules to be used in rule-based MT (Probst et al., 2002; Lavie et al., 2004). Probst et al. (2002) and Lavie et al. (2004) infer transfer rules for MT involving "minor" languages (e.g. Quechua) with very limited resources. To this end, a small parallel corpus (of a few thousand sentences) is built with the help of a small set of bilingual speakers of the two languages. The parallel corpus is obtained by translating a controlled corpus from a "major" language (English or Spanish) to a "minor" language by means of an elicitation tool. This tool is also used to graphically annotate the word alignments between the two sentences. Finally, hierarchical syntactic rules, that can be seen as constituting a context-free transfer grammar, are inferred from the aligned parallel corpus.

On the other hand, in the EBMT framework, some researchers deal with the problem of inferring the kinds of translation rules called *translation templates* (Kaji et al., 1992; Brown, 1999; Cicekli and Guvenir, 2001). A translation template can be defined as a bilingual pair of sentences in which corresponding units (words or phrases) are coupled and replaced by variables. Liu and Zong (2004) provide an interesting review of the different research works dealing with translation templates. Brown (1999) uses a parallel corpus and some linguistic knowledge in the form of equivalence classes (both syntactic and semantic) to perform a generalization over the bilingual examples collected. The method works by replacing each word by its corresponding equivalence class and then using a set of grammar rules to replace patterns of words and tokens by more general tokens. Cicekli and Guvenir (2001) formulate the acquisition of translation templates as a machine learning problem, in which the translation templates are learned from the differences and similarities observed in a set of different translation examples, using no morphological information at all. Kaji et al. (1992) use a bilingual dictionary and a syntactic parser to determine the correspondences between translation units while learning the translation templates. In any case, the translation templates used in EBMT differ from the approach presented in this paper, firstly because our approach is largely based on part-of-speech and inflection information, and the inferred translation rules are flatter, less structured and non-hierarchical (because of this, they are suitable for shallow-transfer MT); and secondly, because the way in which the transformations to

[2]The method is implemented inside package `apertium-transfer-tools` and, released under the GNU GPL license, is freely available at `http://sf.net/projects/apertium`.

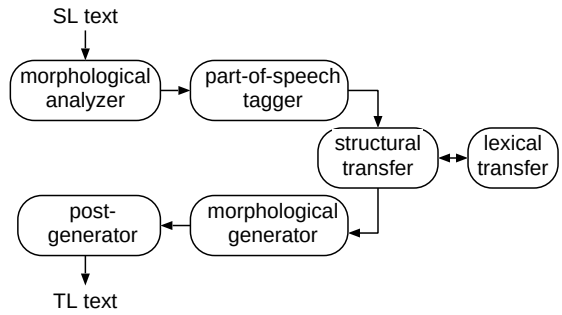


Figure 1: Main modules of the Apertium shallow-transfer MT platform (see section 2). The structural transfer module is the one that applies the inferred transfer rules.

apply are chosen (see section 5) differs from those used in the EBMT framework.

The rest of the paper is organized as follows: the next section overviews the open-source shallow-transfer MT platform Apertium used to test our approach; section 3 overviews the alignment templates (ATs) approach; section 4 explains how to extend the ATs in order to use them to generate (section 5) shallow-transfer rules to be used in MT. Section 6 describes the experiments conducted and the results achieved. Finally, in section 7 we draw some conclusions and outline future work.

2 Overview of Apertium

Apertium[3] (Armentano-Oller et al., 2006) is an open-source platform for developing MT systems, initially intended for related language pairs. The Apertium MT engine follows a shallow transfer approach and may be seen as an assembly line consisting of the following main modules (see figure 1):

A *morphological analyzer* which tokenizes the source-language (SL) text in surface forms and delivers, for each surface form, one or more *lexical forms* consisting of *lemma*, *lexical category* and morphological inflection information.

A *part-of-speech tagger* which chooses, using a first-order hidden Markov model (HMM) (Cutting et al., 1992), one of the lexical forms corresponding to an ambiguous surface form.

A *lexical transfer* module which reads each SL lexical form and delivers the corresponding TL lexical form by looking it up in a bilingual dictionary.

A *structural shallow transfer* module (parallel to the lexical transfer) which uses a finite-state chunker to detect patterns of lexical forms which need to be processed for word reorderings, agreement, etc., and then performs these operations. Note that this is the module that applies the transfer rules generated by the method presented here.

A *morphological generator* which delivers a TL surface form for each TL lexical form, by suitably inflecting it.

A *post-generator* which performs orthographic operations such as contractions (e.g. Spanish *de+el=del*) and apostrophations (e.g. Catalan *el+institut=l'institut*).

Modules use text to communicate, which makes it easy to diagnose or modify the behavior of the system.

2.1 Linguistic data and compilers

The Apertium MT engine is completely independent from the linguistic data used for translating between a particular pair of languages.

Linguistic data is coded using XML-based formats;[4] this allows for interoperability, and for easy data transformation and maintenance. In particular, files coding linguistic data can be automatically generated by third-party tools, as is the case of the method we present.

Apertium provides compilers to convert the linguistic data into the corresponding efficient

[3]The MT platform, documentation, and linguistic data for different language pairs can be freely downloaded from `http://apertium.sf.net`.

[4]The XML (`http://www.w3.org/XML/`) formats for each type of linguistic data are defined through conveniently-designed XML document-type definitions (DTDs) which may be found inside the `apertium` package.

(binary) form used by each module of the engine. Two main compilers are used: one for the four lexical processing modules (morphological analyzer, lexical transfer, morphological generator, and post-generator) and another one for the structural transfer; both generate finite-state processors which make Apertium capable of translating tens of thousands of words per second in a current desktop computer.

3 The alignment templates approach

Alignment templates (ATs) (Och and Ney, 2004), initially used in SMT, perform a generalization over bilingual phrase pairs using word classes instead of words. An AT $z = (S_m, T_n, A)$ consists of a sequence S_m of m SL word classes, a sequence T_n of n TL word classes, and a set of pairs $A = \{(i, j) : i \in [1, n] \wedge j \in [1, m]\}$ with the alignment information between TL and SL word classes.

Learning a set of ATs from a parallel corpus consists of:

1. the computation of the word alignments,
2. the extraction of bilingual phrase pairs, and
3. the substitution of each word by its corresponding word class.

Word alignments. A variety of methods, statistical (Och and Ney, 2003) or heuristic (Caseli et al., 2005), may be used to compute word alignments from a (sentence aligned) parallel corpus. For our experiments (section 6) we have used the open-source `GIZA++` toolkit[5] in the following way. First, standard `GIZA++` training runs to estimate translation models to translate from language L_1 to language L_2, and vice versa. Then, from the training corpus, Viterbi alignments[6] A_1 and A_2 are obtained (one for each translation direction) and symmetrized via the following method (Och and Ney, 2003, p. 33):[7]

> first the intersection $A = A_1 \cap A_2$ of both alignments is computed, then
>
> the alignment A is iteratively extended with alignments $(i, j) \in A_1$ or $(i, j) \in A_2$ if neither SL word $w_{\mathrm{S}j}$ nor TL word $w_{\mathrm{T}i}$ has an alignment in A, or the following two conditions hold:
>
> 1. One of the following (neighboring) alignments $(i \quad 1, j)$, $(i+1, j)$, $(i, j \quad 1)$, $(i, j + 1)$ is already in A.
> 2. The new alignment $A \cup \{(i, j)\}$ does not contain any alignment with both horizontal $((i \quad 1, j), (i + 1, j))$ and vertical $((i, j \quad 1), (i, j + 1))$ neighbors.

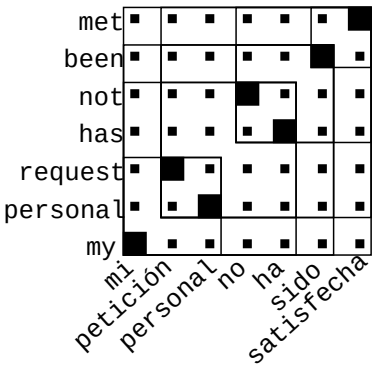


Figure 2: Example of bilingual phrases extracted (see section 3) for a given word-aligned Spanish–English sentence pair in which the alignment information is represented as a binary matrix. Each square corresponds to a bilingual phrase.

Bilingual phrase pairs. The extraction of bilingual phrases (Och et al., 1999) is performed by considering all possible pairs within a certain length and ensuring that (see figure 2):

1. all words are consecutive, and
2. words within the bilingual phrase are not aligned with words from outside.

[5] `http://www.fjoch.com/GIZA++.html`

[6] The Viterbi alignment between source and target sentences is defined as the alignment whose probability is maximal under the translation models previously estimated.

[7] For easier understanding, think about the alignment information as a binary matrix (see figure 2).

The set of bilingual phrases that are extracted from the word-aligned sentence pair $(w_{S1}, \ldots, w_{SJ}), (w_{T1}, \ldots, w_{TI})$ can be formally expressed as follows:

$$BP(w_{S1}^{J}, w_{T1}^{I}, A) = \{(w_{Sj}^{j+m}, w_{Ti}^{i+n}) : \forall (i', j') \in A : j \leq j' \leq j+m \Leftrightarrow i \leq i' \leq i+n\}.$$

Generalization. The generalization is simply done by replacing each word by its corresponding word class. The use of word classes instead of the words themselves allows the description of word reorderings, preposition changes and other divergences between SL and TL.

4 Alignment templates for shallow-transfer machine translation

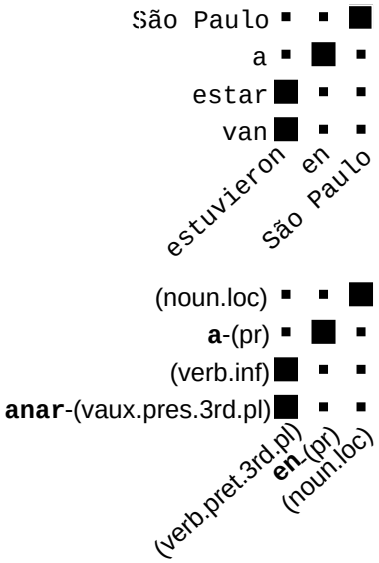
Shallow-transfer MT is an special case of the (indirect) rule-based transfer MT framework. Shallow transfer rules simply detect patterns of lexical forms and apply lexical and syntactic changes to them. Therefore, a simple intermediate representation (IR) consisting of lexical forms is used by the translation engine.

In order for the shallow-transfer MT system to benefit from the AT approach the parallel corpora must be in the same IR used by the translation engine. To that end, the morphological analyzers and part-of-speech taggers of the MT system in which the transfer rules will be applied are used to analyze the parallel corpus before computing the word alignments (see section 3).

4.1 Word-class definition

The transformations to apply are mainly based on the part-of-speech of SL and TL words; therefore, part-of-speech information (including all inflection information such as gender, number or verb tense) is used to define the word class each word belongs to.

Using part-of-speech information to define the set of word classes allows the method to learn syntactic rules such as reordering and agreement rules, and verb tense changes, among others. However, in order to learn lexical changes, such as preposition changes or auxiliary verb usage, additional linguistic information, provided by an expert, is needed.



$$R = \{w_1 = \texttt{verb.*}, w_3 = \texttt{noun.*}\}$$

Figure 3: Example of Spanish–Catalan bilingual phrases (top), alignment template (bottom) obtained when each word is replaced by its corresponding word class, and TL restrictions (see section 4.2) for the Spanish-to-Catalan translation. Words in bold face correspond to lexicalized categories (see section 4.1). Word classes in the horizontal axis correspond to the SL (Spanish) and in the vertical axis to the TL (Catalan). Alignment information is represented as a binary matrix.

Lexicalized categories. A set of (lexicalized) categories usually involved in lexical changes such as prepositions and auxiliary verbs may be provided.[8] For those words whose part-of-speech is in that set of lexicalized categories (from now on, *lexicalized words*) the lemma is also used when defining the word class they belong to. In this way, lexicalized words are placed in single-word classes. For example, if prepositions are considered lexicalized categories, words *to* and *for* would be in different word classes, even if they have the same part-of-speech and inflection information, while words *book* and *house* would be in the same word class (noun, singular). Figure 3 shows an example of Spanish–

[8] Lexicalized categories are specified through a simple XML file.

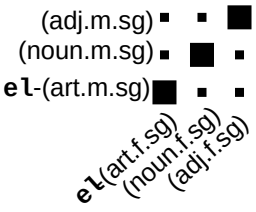
Catalan bilingual phrase and the generalization performed when each word is replaced by its corresponding word class; words in bold face correspond to lexicalized categories. The AT shown in figure 3 generalizes, on the one hand, the use of the auxiliary Catalan verb *anar* to express the past (preterite) tense and, on the other hand, the preposition change when it refers to a place name, such as the name of a city or a country.

4.2 Extending the definition of alignment template

In section 3 an alignment template (AT) was defined as a tuple $z = (S_m, T_n, A)$ in which only the alignment between SL and TL word classes was considered. Here we extend the definition of AT to $z = (S_m, T_n, A, R)$, where a set of restrictions, R, over the TL inflection information of non-lexicalized categories is added.

TL Restrictions. When translating (see next section), that is, when applying ATs, TL inflection information of non-lexicalized words is taken from the corresponding TL word class in the AT being applied, not from the bilingual dictionary; because of this, restrictions are needed in order to prevent an AT to be applied in certain conditions that would produce an incorrect translation. For example, an AT that changes the gender of a noun from masculine to feminine (or vice versa) would produce an incorrect TL word if such a change is not allowed for that noun. Restrictions refer to TL inflection information; therefore, they are obtained for a given translation direction and they change when translating the other way round.

TL restrictions are obtained from the bilingual dictionary. In Apertium bilingual dictionaries, changes in inflection information are explicitly coded. The following two examples show, on the one hand, a Spanish–Catalan bilingual entry and, on the other hand, the restriction over the TL inflection information for the Spanish-to-Catalan translation derived for that bilingual entry:[9]



$$R = \{w_2 = \texttt{noun.m.*}, w_3 = \texttt{adj.*}\}$$

Figure 4: Spanish–Catalan alignment template (AT) and TL restrictions over the inflection information for the Spanish-to-Catalan translation (see section 4.2).

Bilingual entry without any inflection information change

```
<e><p>
 <l>castigo<s n="noun"/></l>
 <r>càstig<s n="noun"/></r>
</p></e>
```

Restriction: w=`noun.*`

Bilingual entry in which the gender changes from feminine (Spanish) to masculine (Catalan)

```
<e><p>
 <l>calle<s n="noun"/>
         <s n="f"/></l>
 <r>carrer<s n="noun"/>
          <s n="m"/></r>
</p></e>
```

Restriction: w=`noun.m.*`

As can be seen, restrictions provide the part-of-speech and inflection information that the lexical form should have at translation time after looking it up in the bilingual dictionary; the star at the end of each restriction means that the rest of inflection information is not restricted. The second bilingual entry would be responsible of the restrictions attached to w_2 in the AT shown in figure 4. That AT generalizes the rule to apply in order to propagate the gender from the noun to the article and the adjective, and can only be applied if the noun (w_2) is masculine in the TL (see next section to know how ATs are applied).

[9] Lemmas between `<l>` and `</l>` XML tags correspond to Spanish words; analogously, lemmas between `<r>` and `</r>` tags correspond to Catalan words. Inflection information is coded through the `<s>` (*symbol*) XML tag, the first one being the part-of-speech.

5 Generation of Apertium structural transfer rules

This section describes the generation of Apertium structural transfer rules; note, however, that the generation of transfer rules for other shallow-transfer MT systems would also be feasible by following the approach presented here.

Apertium structural transfer uses finite-state pattern matching to detect, in the usual left-to-right, longest-match way, fixed-length patterns of lexical forms to process and performs the corresponding transformations. A (generic) shallow-transfer rule consists of a sequence of lexical forms to detect and the transformations to apply to them.

Filtering of the alignment templates. To decide which ATs to take into account for the generation of rules the method is provided with a frequency count threshold. ATs whose frequency count is below this threshold are discarded. In the experiments we have tested two different ways of interpreting the frequency count:

> to use directly the frequency count c, and
>
> to use a modified frequency count $c' = c(1+\log(l))$, where l stands for the length of the SL part of the AT.

The second approach aims at solving the problem caused by the fact that longer ATs have lower frequency counts but may be more accurate as they take more context into account.[10]

Moreover, ATs satisfying one of the following conditions are also discarded:

> the bilingual phrase the AT comes from cannot be reproduced by the MT system in which the transfer rules will be used. This happens when the translation equivalent (in the bilingual dictionary) differs from that in the bilingual phrase extracted from the corpus.
>
> SL and TL non-lexicalized words are not aligned.

Rules generation. In our approach, a rule consists of a set U of ATs with the same sequence of SL word classes, but different sequences of TL word classes, different alignment information or different set of TL restrictions. Formally this may be expressed as follows:

$$U = \{(S_m, T_n, A, R) \in Z : S_m = S^U\},$$

where Z refers to the whole set of extracted ATs and S^U to the sequence of SL word classes all ATs $z \in U$ have in common.

For each set U an Apertium shallow-transfer rule matching the sequence of SL word classes S^U is generated; that rule consists of code applying (see below) always the most frequent AT $z = (S_m, T_n, A, R) \in U$ that satisfies the TL restrictions R. A "default" AT, which translates word for word, is always added with the lowest frequency count. This AT has no TL restrictions and is the one applied when none of the rest can be applied because their TL restrictions are not met.

Code generated for each alignment template. Code is generated by following the order specified by the TL part T_n of the AT. The generated code for each unit in T_n depends on the type of its word class:

> if the word class corresponds to a non-lexicalized word, code is generated to get the translation of the lemma of the aligned SL (non-lexicalized) word by looking it up in the bilingual dictionary, and to attach to the translated lemma the part-of-speech and inflection information provided by the TL word class;
>
> if the word class corresponds to a lexicalized word, it is introduced as is; remember that word classes belonging to lexicalized words store complete lexical forms consisting of lemma, part-of-speech and inflection information.

Note that the information about SL lexicalized words is not taken into account when generating the code for a given AT.

[10] A similar approach was used by Mikheev (1996) in his work on learning part-of-speech guessing rules to prioritize longer suffixes over shorter ones.

Lang.	# sent.	# words
es	100 834	1 952 317
ca	100 834	2 032 925

Table 1: Number of sentences and words in the Spanish–Catalan parallel corpus used for training.

Example of AT application. The following example illustrates how the AT shown in figure 3 would be applied to translate from Spanish to Catalan the input text *vivieron en Francia*.[11] This text segment, after morphological analysis and part-of-speech tagging, is transformed by the MT engine into the intermediate representation *vivir*-(verb.pret.3rd.pl) *en*-(pr) *Francia*-(noun.loc), which becomes the input to the structural transfer module.

The AT is applied in the order specified in its TL part. For the word classes corresponding to non-lexicalized words, the aligned SL words are translated into TL (Catalan) by looking them up in the bilingual dictionary: *vivir* is translated as *viure* and *Francia* is translated as *Franca*. Then, the inflection information provided by the TL part of the AT (see figure 3) is attached to each translated lemma. Finally, word classes corresponding to lexicalized words are just copied to the output as they appear in the TL part of the AT. For the running example the structural transfer output would be: *anar*-(vaux.pres.3rd.pl) *viure*-(verb.inf) *a*-(pr) *Franca*-(noun.loc), which the generation module would transform into the Catalan phrase *van viure a Franca*.

6 Experiments

Task. We have tested our approach on both translation directions of the Spanish–Catalan (es-ca) language pair.[12] Table 1 shows the number of sentences and words in the training parallel corpus; this corpus comes from *El Periodico de Catalunya*,[13] a daily newspaper published both in Catalan and Spanish.

Trans. dir.	Eval. corpus	# words
es-ca	post-edit	10 066
	parallel	13 147
ca-es	post-edit	10 024
	parallel	13 686

Table 2: Number of words of the two different corpora (see section 6) used for evaluation for each translation direction.

The definition of word classes is performed by considering a small set with around 8 lexicalized categories (see section 4.1) for each language. The most common lexicalized categories are: prepositions, pronouns, determiners, subordinate conjunctions, relatives, modal verbs and auxiliary verbs. Remember from section 4.1 that only categories usually involved in lexical changes are lexicalized.

Evaluation. The performance of the presented approach is compared to that of the same MT system when no transfer rules are used at all (word-for-word MT), and that of using hand-coded transfer rules. To that end we calculate the word error rate (WER) computed as the word-level *edit distance* (Levenshtein, 1965) between the translation performed by the MT system for a given setup and a reference translation divided by the number of words in the evaluated translation.

Table 2 shows the number of words of the different corpora used form evaluation. Note that two different evaluation corpora have been used, one (post-edit) in which the reference translation is a post-edited (corrected) version of the MT performed when using hand-coded transfer rules, and another (parallel) in which the text to translate and the reference translation come from a parallel corpus analogous to the one used for training.

Results. Table 3 shows the results achieved for each translation direction and evaluation corpus. The error rates reported are: (a) the results of a word-for-word translation (when no structural transformations are applied),

[11] Translated into English as *They lived in France*.

[12] All linguistic data used can be freely downloaded from http://sourceforge.net/projects/apertium, package apertium-es-ca-1.0.2 .

[13] http://www.elperiodico.com

Trans. dir.	Eval. corpus	No rules	AT count	AT log	Hand
es-ca	post-edit	12.6 %	8.6 %	8.5 %	6.7 %
	parallel	26.6 %	20.4 %	20.4 %	20.8 %
ca-es	post-edit	11.6 %	8.1 %	8.1 %	6.5 %
	parallel	19.3 %	15.0 %	14.9 %	14.5 %

Table 3: Word error rate (WER) for each translation direction and evaluation corpus. The error rates reported are (from left to right): the result when no transfer rules are used, the best result achieved when the count is used directly when discarding infrequent ATs (AT count), the best result achieved when a modified frequency count is used when discarding infrequent ATs (AT log, see section 5), and the results achieved when hand-coded transfer rules are used.

(b) the results when the frequency count is directly used to discard infrequent ATs, (c) the results when a modified frequency count (see section 5) is used to discard infrequent ATs, and (d) the results achieved when using hand-coded transfer rules; in all cases the same linguistic data (morphological and bilingual dictionaries) were used.

As can be seen, when evaluating via a post-edited translation, handcrafted rules perform better than our method; however, they give comparable results when using a evaluation corpus similar to the one used for training. This result suggests, on the one hand, that our training method produces text of the same style of that used for training and, on the other hand, that even though they "learn" the style of the training corpus, the translation quality for other texts is quite good. Note that the post-edited translation used as reference is a corrected version of a MT performed with the same handcrafted rules; therefore, this evaluation is slightly biased towards the system using handcrafted rules.

Finally, note that both criteria used to discard infrequent ATs (see section 5) give comparable results for both translation directions. This may be explained by the fact that, on the one hand, rules that do not apply any AT (because of TL restrictions not being met) perform a word-for-word translation, and on the other hand, rules with longer ATs have more restrictions to check and, therefore, they are more likely to eventually perform a word-for-word translation.

7 Discussion

In this paper the generation of shallow-transfer rules from statistically-inferred alignment templates (ATs) has been tested. To this end, little linguistic information, in addition to the linguistic data used by the MT engine, has been used in order to learn, not only syntactic changes, but also lexical changes to apply when translating SL into TL. This linguistic information consists of a small set of lexical categories involved in lexical changes (prepositions, pronouns, etc.) and can be easily provided.

The method presented has been tested using an existing open-source shallow-transfer MT system. The performance of the system when using the automatically generated rules has been compared to that of a word-for-word translation (when no structural transformations are applied) and that obtained using hand-coded transfer rules. In all cases, there is a significant improvement in the translation quality as compared to word-for-word translation. Furthermore, the translation quality is very close to that achieved when using hand-coded transfer rules, being comparable in some cases.

Finally, we plan to improve the generated rules so that they apply shorter ATs inside the same rule when none of the longer ATs can be applied because of TL restrictions not being met. This gradual "back-off" code in rules would avoid falling back straight into word-for-word translation as it is done now. We also plan to test the presented method with other Apertium-based linguistic packages. Preliminary results on

the Spanish–Portuguese language pair show results in agreement to those provided in this paper when evaluating through a parallel corpus.

Acknowledgements

Work funded by the Spanish Government through projects TIC2003-08681-C02-01 and TIN2006-15071-C03-01, and by the Spanish Government and the European Social Fund through research grant BES-2004-4711. We thank G. Ramırez-Sanchez for her help when defining the set of lexicalized categories.

References

C. Armentano-Oller, R.C. Carrasco, A.M. Corbı-Bellot, M.L. Forcada, M. Ginestı-Rosell, S. Ortiz-Rojas, J.A. Perez-Ortiz, G. Ramırez-Sanchez, F. Sanchez-Martınez, and M.A. Scalco. 2006. Open-source Portuguese-Spanish machine translation. In *Computational Processing of the Portuguese Language, Proceedings of the 7th International Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR 2006*, volume 3960 of *Lecture Notes in Computer Science*, pages 50–59. Springer-Verlag.

D. Arnold, 2003. *Computers and Translation: A translator's guide*, chapter Why translation is difficult for computers, pages 119–142. Benjamins Translation Library. Edited by H. Somers.

R.D. Brown. 1999. Adding linguistic knowledge to a lexical example-based translation system. In *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pages 22–32.

H.M. Caseli, M.G.V. Nunes, and M.L. Forcada. 2005. LIHLA: A lexical aligner based on language-independent heuristics. In *Proceedings of the V Encontro Nacional de Inteligencia Arti cial (ENIA 2005)*, pages 641–650.

I. Cicekli and H.A. Guvenir. 2001. Learning translation templates from bilingual translation examples. *Applied Intelligence*, 15(1):57–76.

D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. 1992. A practical part-of-speech tagger. In *Third Conference on Applied Natural Language Processing. Association for Computational Linguistics. Proceedings of the Conference.*, pages 133–140.

H. Kaji, Y. Kida, and Y. Morimoto. 1992. Learning translation templates from bilingual text. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 672–678. Association for Computational Linguistics.

A. Lavie, K. Probst, E. Peterson, S. Vogel, L. Levin, A. Font-Llitjos, and J. Carbonell. 2004. A trainable transfer-based machine translation approach for languages with limited resources. In *Proceedings of Workshop of the European Association for Machine Translation (EAMT-2004)*.

V.I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848. English translation in Soviet Physics Doklady, 10(8):707-710, 1966.

Y. Liu and C. Zong. 2004. The technical analysis on translation templates. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics (SMC)*, pages 4799–4803. IEEE.

A. Mikheev. 1996. Unsupervised learning of word-category guessing rules. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 327–333.

F.J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F.J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

F.J. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28.

K. Probst, L. Levin, E. Peterson, A. Lavie, and J. Carbonell. 2002. MT for minority languages using elicitation-based learning of syntactic transfer rules. *Machine Translation*, 17(4):245–270.

F. Sanchez-Martınez and H. Ney. 2006. Using alignment templates to infer shallow-transfer machine translation rules. In *Advances in Natural Language Processing, Proceedings of 5th International Conference on Natural Language Processing FinTAL*, volume 4139 of *Lecture Notes in Computer Science*, pages 756–767. Springer-Verlag.

Reordering via N-Best Lists for Spanish-Basque Translation

Germán Sanchis, Francisco Casacuberta
Instituto Tecnológico de Informática
Universidad Politcnica de Valencia
Camino de Vera, s/n. 46022 Valencia, Spain
{gsanchis,fcn}@dsic.upv.es

Abstract

In Statistical Machine Translation (SMT), one of the main problems they are confronted with is the problem stemming from the different word order that different languages imply. Most works addressing this issue centre their effort in pairs of languages involving Arabic, Japanese or Chinese because of their utmost different origin with respect to western languages. However, Basque is also a language with an extremely different word order with respect to most other European languages, linguists being unable to determine its origins with certainty. Hence, SMT systems which do not tackle the reordering problem in any way are mostly unable to yield satisfactory results. In this work, a novel source sentence reordering technique is presented, based on monotonized alignments and n-best lists, endorsed by very promissing results obtained from a Basque-Spanish translation task.

1 Introduction

SMT systems have proved in the last years to be an important alternative to rule-based machine translation systems, being even able of outperforming commercial machine translation systems in the tasks they have been trained on. Moreover, the development effort behind a rule-based machine translation system and an SMT system is dramatically different, the latter being able to adapt to new language pairs with little or no human effort, whenever suitable corpora are available.

The grounds of modern SMT were established in (Brown et al., 1993), where the problem of machine translation was defined as following: given a sentence s from a certain source language, an adequate sentence $\hat{t}$ that maximises the posterior probability is to be found. Such a statement can be specified with the following formula:

$$\hat{t} = \underset{t}{\operatorname{argmax}}\, Pr(t|s)$$

Applying the Bayes theorem on this definition, one can easily reach the next formula

$$\hat{t} = \underset{t}{\operatorname{argmax}}\, \frac{Pr(t) \cdot Pr(s|t)}{Pr(s)}$$

and, since we are maximising over t, the denominator can be neglected, arriving to

$$\hat{t} = \underset{t}{\operatorname{argmax}}\, Pr(t) \cdot Pr(s|t)$$

where $Pr(t|s)$ has been decomposed into two different probabilities: the *statistical language model* of the target language $Pr(t)$ and *the (inverse) translation model* $Pr(s|t)$.

Although it might seem odd to model the probability of the source sentence given the target sentence, this decomposition has a very intuitive interpretation: the translation model $Pr(s|t)$ will capture the word relations

between both input and output language, whereas the language model $Pr(t)$ will ensure that the output sentence is a well-formed sentence belonging to the target language.

In the last years, SMT systems have evolved to become the present state of the art, two of the most representative techniques being the phrase based models (Koehn et al., 2003; Och and Ney, 2004) and the Weighted Finite State Transducers for Machine Translation (Casacuberta and Vidal, 2004; Kumar and Byrne, 2003). Both of these frameworks typically rely on word-aligned corpora, which often lead them to incur in word ordering related errors. Although there have been different efforts aiming towards enabling them to deal with non-monotonicity, the algorithms developed often only account for very limited reorderings, being unable to tackle with the more complex reorderings that e.g. some Asian languages introduce with respect to european languages. Because of this, not only will monotone systems present incorrectly ordered translations, but, in addition, the parameters of such models will be incorrectly estimated, whenever a certain input phrase is erroneously assumed to be the translation of a certain output phrase in training time.

Although no efficient solution has still been found, this problem is well known already since the origin of what is known as statistical machine translation: (Berger et al., 1996) already introduced in their alignment models what they called distortion models, in an effort towards including in their SMT system a solution for the reordering problem. However, these distortion models are usually implemented within the decoding algorithms and imply serious computational problems, leading ultimately to restrictions being applied to the set of possible permutations of the output sentence. Hence, the search performed turns sub-optimal, and an important loss in the representational power of the distortion models takes place.

On the other hand, dealing with arbitrary word reordering and choosing the one which best scores given a translation model has been shown not to be a viable solution, since when allowing all possible word permutations the search is NP-hard (Knight, 1999).

In the present work we develop a new approach to the problem, based on the work of Zens, Matusov and Kanthak (Zens et al., 2004; Matusov et al., 2005; Kanthak et al., 2005), who introduced the idea of monotonizing a corpus. A very preliminary result of our work was published in a Spanish workshop (Sanchis and Casacuberta, 2006). The key idea behind this concept is to use the IBM alignment models to efficiently reorder the input sentence s and produce a new bilingual, monotone pair, composed by the reordered input sentence s' and the output sentence t. Hence, once this new bilingual pair has been produced, the translation model to be applied will not have to tackle with the problems derived from different word reorderings, since this problem will not be present any more. Still, there is one more problem to be solved: in search time, only the input sentence is available, and hence the pair cannot be monotonized. To solve this, a very simple reordering model will be introduced, together with a *reordered* language model and n-best hypothesis generation. In this work, a phrase based model is trained using these monotone pairs.

In the following section, a brief overview of the latest efforts made towards solving the reordering problem will be pointed. In section 3, the approach presented in this work will be described, and in section 4 the experiments performed with this system will be shown. Finally, in section 5 the conclusions from this work will be elucidated, as well as the work that is still to be done.

2 Brief overview of existing approaches

Three main possibilities exist when trying to solve the reordering problem: input sentence reordering, output sentence reordering, or reordering both. The latter is, to the best of our knowledge, as yet unexplored.

Vilar et al. (1996), tried to partially solve the problem by monotonizing the most probable non-monotone alignment patterns and

adding a mark in order to be able to remember the original word order. This being done, a new output language has been defined and a new language and translation model can be trained, making the translation process now monotone.

More recently, Kumar and Byrne (2005) learned weighted finite state transducers accounting for local reorderings of two or three positions. These models were applied to phrase reordering, but the training of the models did not yield statistically significant results with respect to the introduction of the models with fixed probabilities.

When dealing with input sentence reordering (Zens et al., 2004; Matusov et al., 2005; Kanthak et al., 2005), the main idea is to reorder the input sentence in such a way that the translation model will not need to account for possible word reorderings. To achieve this, alignment models are used, in order to establish which word order should be the appropriate for the translation to be monotone, and then the input sentence is reordered in such a manner that the alignment is monotone.

However, this approach has an obvious problem, since the output sentence is not available in search time and the sentence pair cannot be made monotone.

The naïve solution, test on all possible permutations of the input sentence, has already been discussed earlier, being NP-hard (Knight, 1999), as $J!$ possible permutations can be obtained from a sentence of length J. Hence, the search space must be restricted, and such restrictions are bound to yield sub-optimal results. In their work, Kanthak et al. present four types of constraints: IBM, inverse IBM, local and ITG constraints.

Although the restrictions presented in their work (IBM, inverse IBM, local and ITG constraints) did yield interesting results, the search space still remained huge, and the computational price paid for a relatively small benefit was far too high.

- Let:
  - s a source sentence, and s_j its j-th word
  - t a target sentence, and t_i its i-th word
- Let C be a cost matrix
  $c_{ij} = \text{cost}(\text{align}(s_j, t_i))$
- Let $\{s^r\} = \{\text{all possible permutations of } s\}$.
  1. compute alignment $A_D(j) = \underset{i}{\text{argmin}}\, c_{ij}$
  2. $s' = \{s^r | \forall j : A_D(j) \leq A_D(j+1)\}$
  3. recompute (reorder) C, obtaining C'.
  4. set $A'_I(i) = \underset{j}{\text{argmin}}\, c'_{ij}$.
  5. Optional: Compute minimum-cost *monotonic* path through cost matrix C'.

Figure 1: Algorithm for obtaining a monotonic alignment by reordering the source sentence.

3 The reordering model and N-Best reorderings

An important motivation behind the approach in this work is that the reordering constraints presented by Kanthak et al. (Kanthak et al., 2005) do not take into account extremely significant information that can be extracted from monotonized corpora: while reordering the input sentence in such a fashion that the alignment turns monotone, we are performing the reordering step needed further on when this action is needed to be taken on the input test set. Hence, what we would ideally want to do is learn a model using this information that will be capable of reordering a given, unseen, input sentence in the same way that the monotonization procedure would have done, in the hope that the benefits introduced will be greater than the error that an additional model will add into the translation procedure.

Once the alignments made monotonic according to the algorithm shown in Figure 1 (Kanthak et al., 2005), a new source "*language*" has been established, meaning that a reordered language model can be trained with the reordered input sentences s'. Such a language will have the words of the

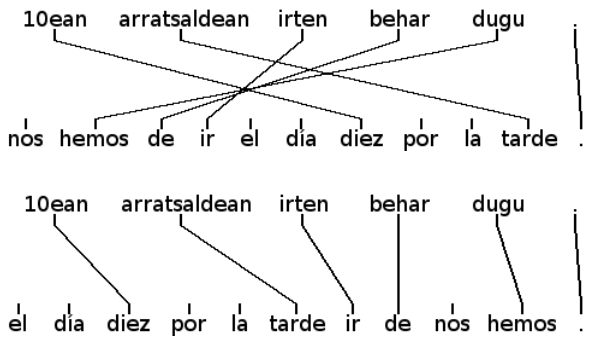


Figure 2: Alignment produced by GIZA (top) and alignment after the monotonization procedure (bottom). This is an example extracted from the Spanish→Basque corpus (i.e. Spanish is the source language). Although these sentences mean "We have to go day 10 in the evening.", the reordered spanish sentence would mean something like "Day ten in the evening go to we have.".

original source language, but the distinctive ordering of the target language. An example of this procedure is shown in Figure 2. Hence, a reordering model can be learnt from the monotonized corpus, which will most likely not depend on the output sentence, whenever the word-by-word translation is accurate enough.

Hence, the reordering problem can be defined as:

$$s' = \underset{s^r}{\operatorname{argmax}}\, Pr(s^r) \cdot Pr(s|s^r)$$

where $Pr(s^r)$ is the reordered language model, and $Pr(s|s^r)$ is the reordering model. Being this problem very similar to the translation problem but with a very constrained translation table, it seems only natural to use the same methods developed to solve the translation problem to face the reordering problem. Hence, in this paper we will be using an exponential model as reordering model, defined as:

$$Pr(s|s') \approx exp(-\sum_i d_i)$$

where d_i is the distance between the last reordered word position and the current candidate position.

		Spanish	Basque
Training	Sentences	38940	
	Different pairs	20318	
	Words	368314	290868
	Vocabulary	722	884
	Average length	9.5	7.5
Test	Sentences	1000	
	Test independent	434	
	Words	9507	7453
	Average length	9.5	7.5

Table 1: Characteristics of the Tourist corpus.

However, and in order to reduce the error that will introduce a reordering model into the system, we found to be very useful to compute an n-best list of reordering hypothesis and translate them all, selecting then as final output sentence the one which obtains the highest probability according to the models $Pr(t) \cdot Pr(s^r|t)$. Ultimately, what we are actually doing with this procedure is to constrain the search space of permutations of the source sentence as well, but taking into account the information that monotonized alignments entail. In addition, this technique implies a much stronger restriction of the search space than previous approaches, reducing significantly the computational effort needed.

4 Translation experiments

4.1 Corpus characteristics

Our system has been tested on a Basque-Spanish translation task, a tough machine translation problem in which reordering plays a crucial role.

The corpus chosen for this experiment is the *Tourist* corpus (Pérez et al., 2005), which is an adaptation of a set of Spanish-German grammars generating bilingual sentence pairs (Vidal, 1997) in such languages. Hence, the corpus is semi-synthetic. In this task, the sentences describe typical human dialogues in the reception desk of a hotel, being mainly extracted from tourist guides. However, because of its design, there is some asymmetry between both languages, and a concept being expressed in several manners

in the source language will always be translated in the same manner in the target language. Because of this, the target language is meant to be simpler than the source language. Since the input language during the design of the corpus was Spanish, the vocabulary size of Basque should be smaller. In spite of this fact, the vocabulary size of Basque is bigger than that of Spanish, and this is due to the agglutinative nature of the Basque language. The corpus has been divided into two separate subsets, a bigger one for training and a smaller one for test. The characteristics of this corpus can be seen in Table 1.

4.2 System evaluation

The SMT system developed has been automatically evaluated by measuring the following rates:

WER *(Word Error Rate)*: The WER criterion computes the minimum number of editions (substitutions, insertions and deletions) needed to convert the translated sentence into the sentence considered ground truth. This measure is because of its nature a pessimistic one, when applied to Machine Translation.

PER *(position-independent WER)*: This criterion is similar to WER, but word order is ignored, accounting for the fact that an acceptable (and even grammatically correct) translation may be produced that differs only in word order.

BLEU *(Bilingual Evaluation Understudy) score*: This score measures the precision of unigrams, bigrams, trigrams, and 4-grams with respect to a set of reference translations, with a penalty for too short sentences (Papineni et al., 2001). BLEU is not an error rate, i.e. the higher the BLEU score, the better.

4.3 Experimental setup and translation results

We used the reordering technique described above to obtain an n-best reordering hypothesis list and translate them, keeping the best scoring one.

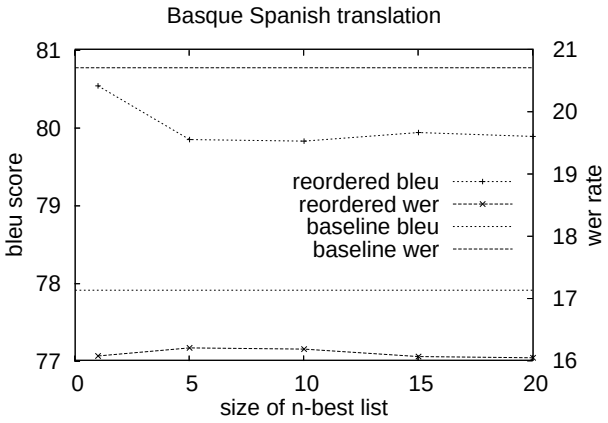


Figure 3: Evolution of translation quality when increasing n for Basque to Spanish.

	Baseline	Reordered, $n = 5$
WER	20.7%	16.2%
BLEU	77.9%	79.8%
PER	12.6%	11.0%

Table 2: Results for Basque to Spanish translation.

First, the bilingual pairs were aligned using IBM model 4 by means of the GIZA++ toolkit (Och and Ney, 2000). After this, the alignments were made monotone in the way described in Figure 1 and a new alignment was recalculated, determining the new monotone alignment between the reordered source sentence and the target, and a reordered source sentence language model was built. In addition, a phrase based model involving reordered source sentences and target sentences was learned by using the Thot toolkit (Ortiz et al., 2005).

For the next step, the reordering model, we used the reordering model built in the toolkit Pharaoh. This was done by including in the translation table only the words contained in the vocabulary of the desired source language, and allowing the toolkit to reorder the words by taking into account the language model and the phrase-reordering model it implements, which is an exponential model. Since in this case, the phrases are just words, what results is an effective implementation of

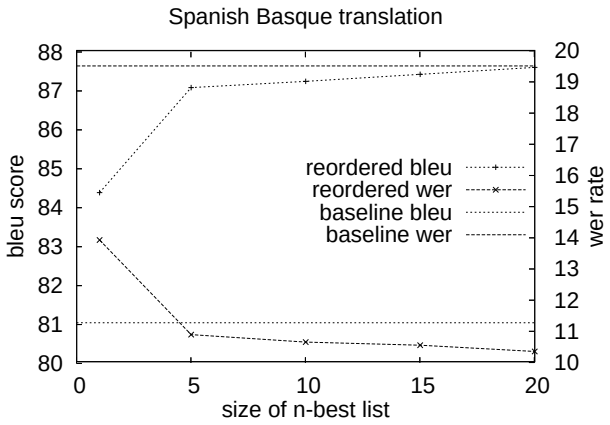


Figure 4: Evolution of translation quality when increasing n for Spanish to Basque.

	Baseline	Reordered, $n = 5$
WER	19.5%	10.9%
BLEU	81.0%	87.1%
PER	6.2%	4.9%

Table 3: Results for Spanish to Basque translation.

an exponential word-reordering model, just as we wanted.

Once the n best reordering hypothesis had been calculated, we translated them all by using Pharaoh once again, and kept the best scoring translation, being the score determined as the product of the (inverse) translation model and the language model.

As a baseline, we took the results of translating the same test set, but without the reordering pipeline, i.e. just using GIZA++ for aligning, Thot for phrase extraction and Pharaoh for translating. The results of this setup can be seen in Table 3 and Table 2, with n-best list size set to 5. At this point, it must be noted that Pharaoh by itself also performs some reordering of the output sentence, but only on a per-phrase basis.

These results show that the reordering pipeline established does have significant benefits on the overall quality of the translation, almost achieving a relative improvement of 50% in WER. Furthermode, it is interesting to point out that even in the case of the PER criterion the results obtained are better. At first sight, this might seem odd, since the PER criterion does not take into account word order errors within a sentence, which is the main problem reordering techniques try to solve. However, this improvement is explained because reordering the source sentence allows for better phrases to be extracted.

It is also interesting to point out that the translation quality when translating from Spanish to Basque is much higher than in the opposite sense. This is due to the corpus characteristics described in the previous section: Spanish being the input language of the corpus, it is only natural that the translation quality will worsen when reversing the meant translation direction. In addition, it can also be observed that the reordering pipeline has less beneficial effects when translating from Basque to Spanish.

Lastly, in Figure 4 and Figure 3, the result of increasing the size of the n-best reordering hypothesis list can be seen. In the case of Spanish-Basque translation, it can be seen how the translation quality still increases until size 20, where as in the case of Basque-Spanish the translation quality already reaches its maximum with the first 5 best hypothesis. However, it can also be seen that just using the best reordering hypothesis already yields better results than without introducing the reordering pipeline. Hence, these figures also show that the phrase extraction process obtains better quality phrases when the monotonization procedure has been implemented before the extraction takes place.

5 Conclusions and Future Work

A reordering technique has been implemented, taking profit of the information that monotonized corpora provide. By doing so, better quality phrases can be extracted and the overall performance of the system improves significantly in the case of a pair of languages with heavy reordering complications.

This technique has been applied to translate a semi-synthetic corpus which deals with the task of Spanish-Basque translation, and

the results obtained prove to be statistically significant and show to be very promising, specially taking into account that Basque is an extremely complex language that poses many problems for state of the art systems.

Moreover, the technique we propose in this paper is learnt automatically, without any need of linguistic annotation or manually specified syntactic reordering rules, which means that out technique can be applied to any language pair without need for any additional development effort.

Both reordered corpora and reordering techniques seem to have a very important potential for the case of very different language pairs, which are the most difficult translation tasks.

As future work, we are planning on obtaining results with other non-synthetic, richer and more complex corpora, as may be other Spanish-Basque corpora or corpora involving language pairs such as Arabic, Chinese or Japanese. In addition, we are planning on developping more specific reordering models, which will be more suitable for this task than the exponential model described here, as well as searching and developing integrated approaches trying to solve the reordering problem.

Acknowledgements

This work has been partially supported by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01 and by the MEC scholarship AP2005-4023.

References

A.L. Berger, P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, J.R. Gillet, A.S. Kehler, and R.L. Mercer. 1996. Language translation apparatus and method of using context-based translation models. In *United States Patent 5510981*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of machine translation. In *Computational Linguistics*, volume 19, pages 263–311, June.

F. Casacuberta and E. Vidal. 2004. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225.

S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney. 2005. Novel reordering approaches in phrase-based statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 167–174, Ann Arbor, Michigan.

K. Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conf. of the NAACL on Human Language Technology*, volume 1, pages 48–54, Edmonton, Canada.

S. Kumar and W. Byrne. 2003. A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proceedings of the 2003 Conference of the NAACL on Human Language Technology*, volume 1, pages 63–70, Edmonton, Canada.

S. Kumar and W. Byrne. 2005. Local phrase reordering models for statistical machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 161–168, Vancouver, Canada.

E. Matusov, S. Kanthak, and H. Ney. 2005. Efficient statistical machine translation with constrained reordering. In *In Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation)*, pages 181–188, Budapest, Hungary.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *ACL 2000*, pages 440–447, Hongkong, China, October.

F.J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(1):417–449.

D. Ortiz, I. Garca-Varea, and F. Casacuberta. 2005. Thot: a toolkit to train phrase-based statistical translation models. In *Tenth Machine Translation Summit*, pages 141–148. Asia-Pacific Association for Machine Translation, Phuket, Thailand, September.

Papineni, A. Kishore, S. Roukos, T. Ward, and W. Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. In *Technical Report RC22176 (W0109-022)*, IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY.

A. Pérez, F. Casacuberta, M.I. Torres, and V. Guijarrubia. 2005. Finite-state transducers based on k-tss grammars for speech translation. In *Proceedings of Finite-State Methods and Natural Language Processing (FSMNLP 2005)*, pages 270–272, Helsinki, Finland, September.

G. Sanchis and F. Casacuberta. 2006. N-best reordering in statistical machine translation. In *IV Jornadas en Tecnología del Habla*, pages 99–104, Zaragoza, Spain, November.

E. Vidal. 1997. Finite-state speech-to-speech translation. In *Proceedings of ICASSP-97*, volume 1, pages 111–114, Munich, Germany.

J.M. Vilar, E. Vidal, and J.C. Amengual. 1996. Learning extended finite-state models for language translation. In *Proc. of Extended Finite State Models Workshop (of ECAI'96)*, pages 92–96, Budapest, August.

R. Zens, H. Ney, T. Watanabe, and E. Sumita. 2004. Reordering constraints for phrase-based statistical machine translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pages 205–211, Geneva, Switzerland.

Demonstration of the Greek to English METIS-II MT System

Sofianopoulos Sokratis
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str., 151 25
Paradissos Amaroussiou Athens, Greece
s_sofian@ilsp.gr

Spilioti Vassiliki
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str., 151 25
Paradissos Amaroussiou Athens, Greece
vspiliot@ilsp.gr

Vassiliou Marina
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str.,
151 25 Paradissos Amaroussiou
Athens, Greece
mvas@ilsp.gr

Yannoutsou Olga
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str.,
151 25 Paradissos Amaroussiou
Athens, Greece
olga@ilsp.gr

Markantonatou Stella
Institute for Language & Speech Processing
6 Artemidos & Epidavrou Str.,
151 25 Paradissos Amaroussiou
Athens, Greece
marks@ilsp.gr

Abstract

METIS-II, the MT system presented in this paper, does not view translation as a transfer process between a source language (SL) and a target one (TL), but rather as a matching procedure of patterns within a language pair. More specifically, translation is considered to be an assignment problem, i.e. a problem of discovering each time the best matching patterns between SL and TL, which the system is called to solve by employing pattern-matching techniques.

Most importantly, however, METIS-II is innovative because it does not need bilingual corpora for the translation process, but exclusively relies on monolingual corpora of the target language.

1 Introduction

The system presented here further elaborates on the original METIS approach (Dologlou et al., 2003) which did not view translation as a transfer process between a source language and a target one, but rather as a matching procedure of patterns within a language pair (Markantonatou et al., 2006). With this approach, only basic NLP resources (such as taggers, lemmatisers, chunkers and simple bilingual lexica) are needed, while new languages, especially low density ones, can be easily included in the system. Furthermore, bilingual corpora are no longer essential; monolingual corpora of the target language suffice for the translation process.

METIS-II extends the original idea by handling patterns (translation units) at sub-sentential level, thus facilitating the elicitation of linguistic information from the TL corpus such as syntactic and/or semantic preferences of words as well as word order.

Four language pairs have been developed as yet, namely Dutch, German, Greek and Spanish into English, all with satisfactory results in terms of BLEU (Papineni et al. 2002) and NIST (2002) evaluations (Tambouratzis et al., (2006) and METIS II – Deliverable 5.2 (2007)); however, the METIS-II system reported here concerns only the Greek into English language pair.

METIS-II system comprises roughly four (4) modules/phases, namely Pre-processing (transformation of the input into patterns), Core Engine (pattern matching), Token Generation (creation of word forms) and Synthesising (composition of the final translation).

The structure of the paper is as follows: in Section 2 the main system features are presented. Sections 3, 4, 5 and 6 describe the respective system modules. Section 7 reports on system testing and

evaluation results; section 8 provides a brief description of the translation process, while the last section summarises the plans for the future development and optimisation of the system.

2 System Features

METIS-II is regarded to be of hybrid nature, since it joins pattern-matching techniques with statistical information, while employing algorithms for handling combinatorial optimisation problems (such as the assignment problem). In addition, a very limited number of linguistic rules is employed, thus avoiding the explosion of rules in rule-based grammars (Gaizauskas, 1995).

Moreover, within this system, what is crucial is the notion of patterns, that is, phrasal models (tokens, chunks, clauses, sentences), which form the basis for measuring the similarity between SL and TL. Patterns are generated by the tools used for both languages and differ from the patterns employed in the corpus-based MT paradigm mostly in the sense that they are viewed as models of TL strings, which receive their final form after corpus consultation.

Therefore, METIS II is different both at implementation level, given that it employs a variety of algorithms, and conceptually, since translation is viewed as a matching process of patterns between SL and TL, aiming each time at detecting the best match.

Nowadays investigation of hybrid systems combines easy-to-obtain resources from all MT paradigms and shows a very promising path in research (Thurmair, 2005).

3 Pre-Processing

For the translation process both the SL input and the TL corpus are transformed to sets of patterns, which are generated with standard NLP techniques.

3.1 TL pattern generation

The TL pattern generation involves the off-line pre-processing of the British National Corpus (BNC[1]), which has been selected as TL corpus. BNC pre-processing comprises the following steps:

- Lemmatisation with a reversible lemmatiser (Carl et al., 2005)
- Segmentation of text into finite clauses with a purpose-built tool
- Syntactic annotation at chunk level with ShaRPa 2.0 chunker (Vandeghinste, 2005)
- Corpus indexation to allow for an efficient and fast search for a best match: in particular, clauses are indexed according to their finite verb, while chunks are classified according to their labels.

3.2 SL pattern generation

The SL pattern generation involves the annotation of the SL input by a tokeniser, lemmatiser, tagger (Labropoulou et al., 1996) and a chunker (Boutsis et al., 2000), resulting in a sequence of labelled patterns[2] and their contained tokens. In addition, the respective heads are identified.

This sequence is then enhanced by the Lexicon look-up, which provides all the possible translation equivalents together with PoS information, resembling thus a TL sequence.

It should be noted that the METIS-II system receives as input a sequence of sentences, but it handles each contained clause separately, synthesising in the end the translations of the various segments.

4 Core Engine

The core engine of METIS-II system is fed with a sequence of TL-like patterns (created as described in Section 3.2), which is handled by the pattern-matching algorithm in order to produce the final translation.

A characteristic feature of the pattern-matching algorithm, which mimics and exploits the recursive nature of language, is that it proceeds in stages: moving from wider patterns to narrower ones, it manages to discover the longest similar pattern in terms of overall structure and lexical head affiliations and then identify and correct any residual mismatches. Similarity is calculated on the basis of a series of weights, which mainly reflect grammatical information.

More specifically, the system searches the TL corpus for candidate patterns of clauses, which are similar to the given TL-like clause pattern in terms

[1] www.natcorp.ox.ac.uk/

[2] The pattern labels denote the categorical status of patterns.

of the main verb and the number of contained chunk patterns (Step 1).

In accordance to the above, the first comparison is performed at clause level, where similarity is calculated on the basis of the main verb, the chunk labels and the head lemmas, resulting in the establishment of chunk order within the TL-like clause (Step 2).

The subsequent comparison is narrower and confined within the boundaries of the chunk patterns. The pattern-matching algorithm calculates the similarity of contained tokens, fixing thus the correct order of tokens within each chunk (Step 3).

At the end of the comparison process a TL corpus clause is selected as the basis of translation, while chunk and token order has been established. Nevertheless, the final translation is derived from the specific corpus clause, only after the contained chunks have been processed, with the purpose of eliminating any mismatches. This processing entails either modification or substitution of given chunks, in order to include them in the final translation (Step 4).

The output of the pattern-matching algorithm is a sequence of translated lemmas and their respective tags, which is subsequently fed into the token generation module.

5 Token Generation

The token generation module receives as input a sequence of translated lemmas and their respective tags and is responsible for the production of word forms (tokens) out of lemmas and the handling of agreement phenomena, for instance subject-verb agreement, on the basis of morphological information.

For the generation task, METIS-II utilises resources produced and used in the reversible lemmatiser/token-generator for English (Carl et al., 2005).

The morphological features identified and used, which are essential for the specific TL, namely English, are tense, person, number, case and degrees of comparison (comparative and superlative degree). These features are integrated within the inflection rules employed for token generation.

Furthermore, morphological information is exploited for handling the syntactic phenomenon of subject-verb agreement, especially in cases of an empty subject. Given that Greek is a pro-drop language, subjectless clauses often occur. The generation module is based on the morphological features of the main verb of a given clause, in order to derive a suitable subject pronoun on every occasion.

6 Synthesising

As mentioned above, METIS-II receives as input a text, i.e. a sequence of sentences. Sentences consist of clauses, and very often a clause may be discontinued through the embedding of another clause. The METIS-II core engine creates separate translation processes for each clause, namely each clause process is a separate thread, running in parallel with the others. When a clause thread has finished translating, it reports back to the core engine.

When all SL clause processes have reported back, the corresponding target sentence is formed. Clauses are placed in the target sentence in the same order as they are found in the source sentence. However, in cases of discontinuous embedding, the translation output consists of clauses placed next to each other.

When the synthesising phase is concluded for a given sentence, then this sentence is added to the final text, following source text sentence order.

The entire translation process, from the input of the TL-like pattern to the core engine up to the synthesising phase, is presented in Figure 1.

7 System Testing and Evaluation

In the present section the results obtained for the Greek → English language pair are summarised. The experiment involved testing METIS-II in comparison to SYSTRAN, a commercial, widely-used MT system, which is mainly rule-based.

7.1 Experimental set-up

The corpus tested was extracted from real texts, mainly from newspapers, and consisted of fifty (50) sentences. The test sentences had an average length of 8,2 words, were of relative complexity, containing one to two clauses each and covered various syntactic phenomena such as word-order variation, NP structure, negation, modification etc.

There was no limitation defined regarding the possible translations of each source token, while the reference translations used for the evaluation have been restricted to three (3) and were produced by humans.

With respect to the evaluation of both MT systems, METIS-II and SYSTRAN, established metrics in the MT field were employed, namely BLEU (Papineni et al. 2002) and NIST (2002), which rely on calculating matching n-grams over words, as well as the Translation Error Rate (TER), which measures the amount of editing that a human would have to perform to change a system output, so that it exactly matches a reference translation (Snover et al., 2006: 1).

7.2 Experimental results

The experimental results obtained are summarised in Tables 1-3, where the mean of the 50 sentence scores obtained for each system are indicated, together with the median, the standard deviation, as well as the maximum and minimum scores.

As can be seen from Table 1, where the evaluation results based on the BLEU metric are presented, both systems exhibit the same maximum and minimum accuracy; however, METIS-II has a significantly higher mean accuracy. More specifically, METIS-II achieves perfect scores for 16% of the test sentences, while the respective SYSTRAN percentage is 4%.

Nevertheless, SYSTRAN gets slightly better scores at the middle score range, which explains why this system has a higher median accuracy. Moreover, SYSTRAN seems to be more stable, given that its scores are characterised by a lower standard deviation.

With respect to the NIST metric, the picture seems more straightforward. METIS-II consistently generates more accurate translations, while SYSTRAN continues behaving in a more stable manner, since its standard deviation is lower.

The opposite conclusions are obtained, as regards the TER metric, according to which the lowest scores are equated to a smaller number of edits. Therefore, apart from its high maximum accuracy, SYSTRAN consistently exhibits a better mean and median accuracy, while once more is proved to be a more stable system than METIS-II, since its scores are characterised again by a lower standard deviation. It should be noted, though, that METIS-II achieved a perfect translation for 9 out of the 50 sentences, while SYSTRAN translated perfectly only 3.

In order to investigate whether the differences in the accuracy populations (where each sentence corresponds to one element of the population) of the two systems, METIS-II and SYSTRAN, are significant, a set of t-tests were performed on the metric (BLEU, NIST, TER) results per system. More specifically, 3 paired t-tests were performed, in order to determine whether the means of the translation scores for the two systems differed significantly.

The output of the t-tests indicated that the differences in the mean accuracy of the two systems were not statistically significant for any of the three metrics at a confidence level of 95%.

	METIS-II	**SYSTRAN**
Mean accuracy	0,3841	0,3214
Median accuracy	0,3537	0,3715
Standard Deviation	0,3718	0,2960
Maximum accuracy	1,0000	1,0000
Minimum accuracy	0,0000	0,0000

Table 1. Comparative analysis of the sentence results for METIS-II and SYSTRAN using the **BLEU** metric

	METIS-II	**SYSTRAN**
Mean accuracy	6,8088	6,3128
Median accuracy	7,4175	6,6791
Standard Deviation	2,5878	2,2869
Maximum accuracy	10,9051	10,8134
Minimum accuracy	1,2651	0,4828

Table 2. Comparative analysis of the sentence results for METIS-II and SYSTRAN using the **NIST** metric

	METIS-II	**SYSTRAN**
Mean accuracy	33,7873	33,3587
Median accuracy	34,7700	29,2855
Standard Deviation	23,9438	21,1764
Maximum accuracy	90,9090	105,8820
Minimum accuracy	0,0000	0,0000

Table 3. Comparative analysis of the sentence results for METIS-II and SYSTRAN using the **TER** metric

8 Web Application

METIS-II has been implemented as a web application, providing a common interface (Figure 2) for all four language pairs. The whole process is pretty simple, with the end user selecting the preferred source language and entering a sentence for translation. When the “Translate” button is pressed, the corresponding web service is initiated and the given sentence is handled by the various system modules.

When the translation process is terminated, the result appears on the web page, while the intermediate system outputs are available to the end user in .html form (Figure 3).

9 Future plans

In METIS-II we have succeeded in restricting the use of structure-modifying rules by using adjustable weights in various phases of the translation process. The employment of adjustable weights makes it possible for the system to move within, i.e. to choose from, a range of potential decisions, thus leading to a different translation output.

Apart from delimiting the use of rules, weights also render METIS-II user-customisable, as the system can be tuned to the end user needs via appropriate weight selection. In this way, the system adapts to a specific operational environment and the output gradually improves, leaving intact the processes of the core engine.

At this point of development, however, all the aforementioned weights have been initialised manually, based on intuitive knowledge. What is, thus, essential is an automated process for defining and calculating the optimal weight values. To achieve that, exploration of appropriate machine learning methods has been planned.

References

Boutsis, S., Prokopidis, P., Giouli, V. & Piperidis, S. (2000). A Robust Parser for Unrestricted Greek Text. In Proceedings of the Second International Conference on Language Resources and Evaluation, Vol. 1 (pp. 467—482). Athens, Greece.

Carl, M., Schmidt, P. & Schütz, J. (2005). Reversible Template-based Shake & Bake Generation. In Proceedings of the Example-Based Machine Translation Workshop held in conjunction with the 10th Machine Translation Summit (pp. 17--26). Phuket, Thailand.

Dologlou, I., Markantonatou, S., Tambouratzis, G., Yannoutsou, O., Fourla, A. & Ioannou, N. (2003). Using Monolingual Corpora for Statistical Machine Translation. In Proceedings of EAMT/CLAW 2003 (pp. 61--68). Dublin, Ireland.

Gaizauskas, R. (1995). Investigations into the Grammar Underlying the Penn Treebank II. Research Memorandum CS-95-25, Department of Computer Science, University of Sheffield.

Labropoulou, P., Mantzari, E. & Gavrilidou, M. (1996). Lexicon-Morphosyntactic Specifications: Language-Specific Instantiation (Greek), PP-PAROLE, MLAP 63-386 report.

Markantonatou, S., Sofianopoulos, S., Spilioti, V., Tambouratzis, G., Vassiliou, M. & Yannoutsou, O. (2006). Using Patterns for Machine Translation (MT). In Proceedings of the European Association for Machine Translation (pp. 239--246). Oslo, Norway.

METIS II – Deliverable 5.2 (2007). Validation & Fine-Tuning Results for the First Prototype.

NIST (2002). Automatic Evaluation of Machine Translation Quality Using n-gram Co-occurrences Statistics (Available at www.nist.gov/speech/tests/mt/)

Papineni, K.A., Roukos, S., Ward, T. & Zhu, W.J. (2002). BLEU: A method for automatic evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (pp. 311--318). Philadelphia, USA.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L. & Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of Association for Machine Translation in the Americas.

Tambouratzis, G., Sofianopoulos, S., Spilioti, V., Vassiliou, M., Yannoutsou, O. & Markantonatou S. (2006). Pattern matching-based system for Machine Translation (MT). In Proceedings of Advances in Artificial Intelligence: 4th Hellenic Conference on AI, SETN 2006, Vol. 3955 (pp. 345—355). Heraklion, Crete, Greece. Lecture Notes in Computer Science, Springer Verlag.

Thurmair, G. (2005). Improving MT Quality: Towards a Hybrid MT Architecture in the Linguatec 'Personal Translator'. Talk given at the 10th MT Summit. Phuket, Thailand

Vandeghinste, V. (2005). Manual for ShaRPa 2.0. Internal Report. Centre for Computational Linguistics, K.U.Leuven.

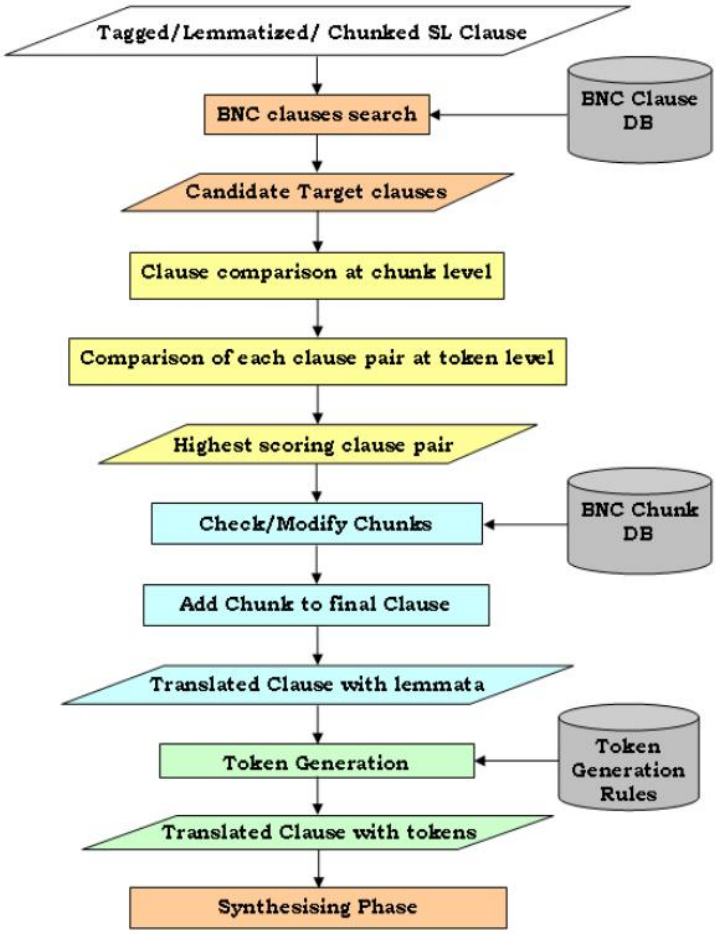


Figure 1: Clause Processing

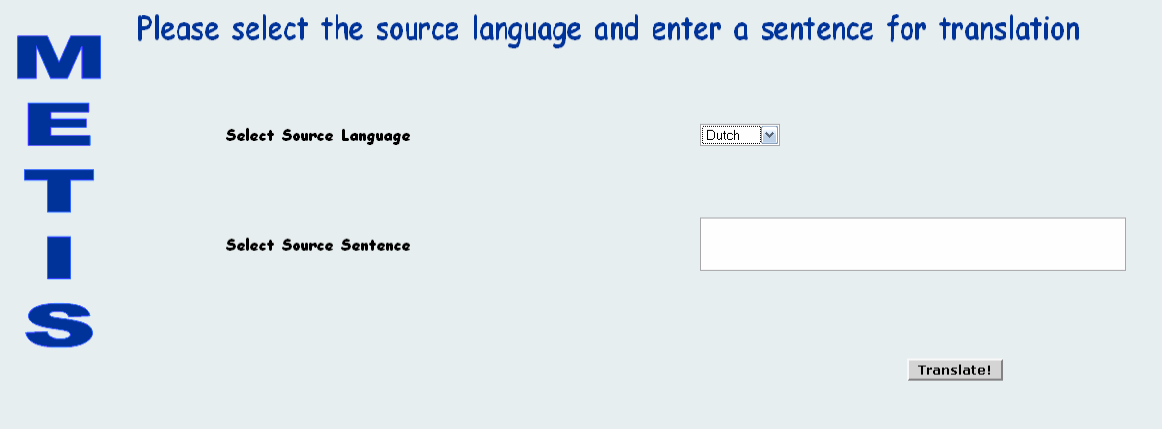


Figure 2: METIS-II home page

Source Clause: pp(np_nm(the greece**)) vg(**dismiss|fail|reject**) ppof(pp(np_ac(**any|anyone|each|every face|figure|form**)) pp(np_ge(**xenophobia**)))**

Final Clause: Greece rejects every form of xenophobia

Source Chunk:	pp([-{-}] np_nm(the{at} [greece{np0}]))
Corpus Chunk:	pp([-{-}] np_1([piatakov{NP0}]))
Final Chunk:	**pp([-{-}] np_1([greece{np0}]))**

Score=82.85715%	-{-}	the{at}	greece{np0}
-{-}	**-{-} 100.0%**	the{at} 0.0%	greece{np0} 0.0%
piatakov{NP0}	-{-} 0.0%	the{at} 0.0%	**greece{np0} 80.0%**
PAD	null 20.0%	**null 20.0%**	null 20.0%

- Processing Corpus chunk: pp([-{-}] np_1([Piatakov]))
- Replacing [piatakov{NP0}] with token:greece{np0}
- NOT Adding the{at}

Source Chunk:	vg([reject{vv}])
Corpus Chunk:	vg([reject{VVD}])
Final Chunk:	**vg([reject{vv}])**

Score=100.0%	reject{vv}
reject{VVD}	**reject{vv} 100.0%**

- Keeping chunk :vg([reject{vv}])

Source Chunk:	ppof(pp([-{-}] np_ac(any{dt0}|anyone{pn}|each{pn}|every{at} [face{nn}|figure{nn}|form{nn}])) pp([-{-}] np_ge([xenophobia{nn}])))
Corpus Chunk:	ppof(pp([as{PRP}] np_2(a{AT0} [form{NN1}])) pp([of{PRF}] np_2([dazzle{NN1}])))
Final Chunk:	**ppof(pp([-{-}] np_2(every{at} [form{nn}])) pp([of{PRF}] np_2([xenophobia{nn}])))**

Score=70.0%	-{-}	any{dt0}|anyone{pn}|each{pn}|every{at}	face{nn}|figure{nn}|form{nn}	-{-}	xenophobia{nn}
as{PRP}	**-{-} 70.0%**	any{dt0}!anyone{pn}!each{pn}!every{at} 0.0%	face{nn}!figure{nn}!form{nn} 0.0%	-{-} 70.0%	xenophobia{nn} 0.0%
a{AT0}	-{-} 0.0%	**every{at} 70.0%**	face{nn}!figure{nn}!form{nn} 0.0%	-{-} 0.0%	xenophobia{nn} 0.0%
form{NN1}	-{-} 0.0%	any{dt0}!anyone{pn}!each{pn}!every{at} 0.0%	**form{nn} 100.0%**	-{-} 0.0%	xenophobia{nn} 40.0%
of{PRF}	-{-} 70.0%	any{dt0}!anyone{pn}!each{pn}!every{at} 0.0%	face{nn}!figure{nn}!form{nn} 0.0%	**-{-} 70.0%**	xenophobia{nn} 0.0%
dazzle{NN1}	-{-} 0.0%	any{dt0}!anyone{pn}!each{pn}!every{at} 0.0%	face{nn}!figure{nn}!form{nn} 40.0%	-{-} 0.0%	**xenophobia{nn} 40.0%**

- Replacing [as{PRP}] with token:-{-}
- Replacing [a{AT0}] with token:every{at}
- Replacing [form{NN1}] with token:form{nn}
- Replacing [of{PRF}] with token:of{PRF}
- Replacing [dazzle{NN1}] with token:xenophobia{nn}

Figure 3: Step 4 output

Theoretical and methodological issues regarding the use of Language Technologies for patients with limited English proficiency

Harold Somers
School of Languages, Linguistics and Cultures
University of Manchester
Oxford Road, Manchester M13 9PL
Harold.Somers@manchester.ac.uk

Abstract

This paper concerns the use of spoken language translation as well as other technologies to support communication between clinicians and patients where the latter have limited proficiency in the majority language. The paper explores some theoretical and methodological issues, in particular the question of whether it is the patient or clinician who should be seen as the primary user of such software, and whether for certain scenarios more simple technology is preferable, especially given the huge overheads involved in developing SLT systems for under-resourced languages. A range of solutions are discussed.

1 Introduction

As its title suggests, this paper seeks to explore issues around the problem of using language technologies to support patients and healthcare providers where there is a significant language barrier. For convenience, in the title and elsewhere we use the phrase "patients with limited English proficiency (LEP)", though it should be understood of course that much of the discussion would apply equally to other countries where the host or majority language is another language.

This paper is essentially theoretical and methodological, and although it does incorporate reflections on some recently completed pieces of research, it should be understood chiefly as a statement of the author's views, and if it is in some respects confrontational or controversial, then this is in a sense deliberate.

In any western country there are recent or long-term immigrants, refugees, and asylum seekers and other people whose command of English, while often adequate for day-to-day activities such as shopping and other domestic chores, is not sufficient for more formal situations such as interactions with health services, especially visits to their doctor. There is no shortage of literature reporting disparities in healthcare provision in these communities and communication difficulties are identified as a major factor (e.g. Jones & Gill 1998, Fassil 2000, Jacobs et al. 2001, Bischoff et al. 2003, Flores et al. 2005, Westberg & Sorensen 2005), and an equally rich literature, which we will not review here, discusses traditional ways of addressing this problem, through use of interpreters and other services. Our interest is in the extent that language technology, including but not limited to machine translation (MT), may be able to provide some support as a contribution to a solution to this problem (Somers & Lovel 2003).

Two aspects of this issue need to be underlined immediately. First, it should be realised that this is a problem not just for the LEP patients, but for the healthcare providers with whom they need to interact: it is a matter not only of making oneself understood, but of understanding too. This seems to be an obvious point, but is often overlooked, for example in papers with titles referring to "problems of refugees" and so on, when more properly the focus should be on "problems of communication". By the same token, note the use of the term "healthcare providers": this is not just a problem for doctors, but for a wide range of professionals with whom patients must interact on the pathway to healthcare.

This brings us to the second point: while it is natural to focus on the doctor–patient consultation as the central element of the "pathway to healthcare", in fact, this is only one of many diverse

interactions that a patient has with a variety of healthcare providers, including receptionists at clinics and hospitals, paramedics, nurses, therapists, pharmacists as well, of course, as the "doctor" who may be a GP, a consultant, a specialist, and so on. Each of these interactions involves a range of communicative activities requiring different language skills and implying different language technologies, often but not inevitably involving *translation* in some form.

In this paper we will first explore this issue of different users and different scenarios, always focusing on how particular aspects of this impact on the choice and design of language technology.

We will then look in particular at the doctor–patient interview and compare the relatively sophisticated approach of using Spoken Language Translation (SLT) as compared to use of much simpler technology, as tested in some recent research by the present author.

2 Different users, different scenarios

As stated above, although it is natural to think of "going to the doctor" as involving chiefly an interview with a GP, and while everything in medical practice arguably derives from this consultation, the pathway to healthcare in normal circumstances involves several other processes, all of which involve language-based encounters that present a barrier to LEP patients. Let us consider the range of processes, interlocutors, and possible technologies that might be suitable, reiterating some points made previously by this author (Somers 2006).

2.1 The pathway to healthcare

The pathway might begin with a person suspecting that there may be something wrong with them. Many people nowadays would in this situation first try to find out something about their condition on their own, typically on the Word-Wide Web. If you need this information in your own language, and you have limited literacy skills, as is the case with many asylum seekers and refugees, technologies implied are multilingual information extraction, MT perhaps coupled with text simplification, with synthesized speech output. For specific conditions which may be treated at specialist clinics (our own experience is based on Somalis with respiratory difficulties) it may be possible to identify a series of frequently asked questions and set up a pre-consultation computer-mediated help-desk and interview (cf. Osman et al. 1994).

Having decided that a visit to the doctor is indicated, the next step is to make an appointment. Appointment scheduling is the classical application of SLT, as seen in most of the early work in the field, and is a typical case of a task-oriented cooperative dialogue. Note that the dialogue partner – the receptionist in the clinic – does not necessarily have any medical expertise, nor possibly the high level of education and openness to new technology that is often assumed in the literature on SLT.

If this is the patient's first encounter with this particular healthcare institution, they may wish to get their "history", a task nowadays often done separately from the main doctor–patient consultation, to save the doctor's time. This might be a suitable application for computer-based interviewing (cf. Bachman 2003).

The next step might be the doctor–patient consultation itself, which has been the focus of much attention (e.g. papers at the recent *Workshop on Medical Speech Translation* at HLT/NAACL in New York in 2006). While some developers (e.g. Bouillon et al. 2005) originally assumed that the patient's role in this can be reduced to simple responses involving yes/no responses, gestures and perhaps a limited vocabulary of simple answers, current clinical theory in contrast focuses on *patient*-centred medicine (cf. Stewart et al. 2003), an approach now adopted by Bouillon et al. (2007). The session will see the doctor eliciting information in order to make a diagnosis as foreseen, but also explaining the condition and the treatment, exploring the patient's feelings about the situation, and inviting the patient to ask questions. So the dialogue is very much a two-way interaction. Of course this presents massive difficulties for SLT system design.

After the initial consultation, the next step may involve a trip to the pharmacist to get some drugs or equipment. Apart from the human interaction, the drugs (or whatever) will include written instructions and information: frequency and amount of use, contraindications, warnings and so on. This is an obvious application for controlled language MT: drug dose instructions are of the same order of complexity as weather bulletins, though there remains the practical problem of

transferring the text from the packet to the translation system. For non-literate patients, "talking pill boxes" are already available (marketed by MedivoxRx, see Orlovsky 2005), so it would be nice if they could "talk" in a variety of languages.

Another outcome might involve another practitioner – a nurse or a therapist – and a series of meetings where the condition may be treated or managed. Apart from more scheduling, this will almost certainly involve explanations and demonstrations by the practitioner, and typically also elicitation of further information from the patient. Hospital treatment would involve interaction with a wide range of staff, again not all medical experts.

All this introduces the question of who is the principle user of a communication device, which will have a bearing on many design issues. In contrast for example with several medical SLT designs, where it is assumed that the doctor is the one who controls the dialogue and accordingly controls the SLT system interface (Narayanan et al. 2004:101, Bouillon et al. 2005, Ettelaie et al. 2005:89), we might propose that it is the *patient* who is going to be the regular user, and who should therefore "own" the device.

At the very least, it should be recognised that a communication device (whether SLT or some other technology, see below) will typically have two users at any time, who may have very different skills and expectations, and these need to be taken into consideration in the design. Indeed, just like the healthcare providers, as already mentioned, not all patients are alike, and they may represent a wide range of levels of language ability (both native and target), literacy, computer literacy, and a variety of expectations and experiences regarding healthcare itself. It is therefore obvious that interfaces to any communication systems should be flexible, and possibly different depending on the profile of the user.

Realisitically, we are not going to address all these problems, but let us consider some of the basic technology issues that the different usage scenarios introduce.

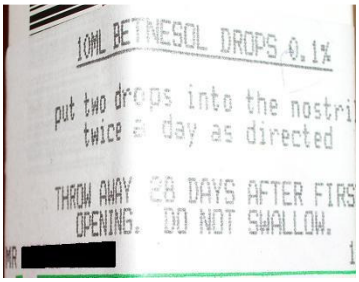
2.2 Language technology implications

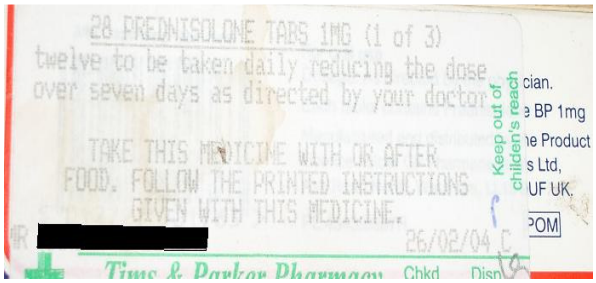
Our discussion so far has mentioned or implied a number of basic technologies including SLT, text MT, multilingual information extraction, text simplification, and computer-based interviewing, automatic speech recognition (ASR) and speech synthesis. Let us focus on applications involving translation.

One obvious problem for these technologies is that often the language we are interested is one of the so-called "under-resourced" languages: this severely limits what can be done, and precludes for example using off-the-shelf components, since they simply do not exist. The effort required to develop SLT for an under-resourced language should not be underestimated (cf. Black et al. 2002, Schultz et al. 2004, Zhou et al. 2004, Narayanan et al. 2004, 2006, Kathol et al. 2005, Besacier et al. 2006, Schultz & Black 2006). We have explored the possibility of "faking" speech synthesis as an interim solution to this (Evans et al. 2002, Somers et al. 2006) with a fairly promising evaluation based on the doctor–patient dialogue scenario using a German synthesizer to produce fake Somali output. Currently we are attempting the more audacious task of "fake" speech recognition by tricking an English ASR system into recognizing a limited vocabulary of Urdu words, with astonishingly good results when the system has to choose from a set of possible responses (Rizvi, in prep.).

Even with languages that are better resourced, developing applications suitable for this scenario can be challenging. For example, Wang (2007) reports a Chinese–English SLT system built by pipelining commercially available Chinese ASR, Chinese–English MT and English speech synthesis, tested once again in the healthcare scenario. Replicating the evaluation methodology of Somers & Sugita (2003) in which subjects are asked to identify the intended meaning of a translated answer to a specific question, he found that Chinese ASR is the weakest link in the chain with around 70% correct interpretation of ASR+MT, dropping to 62% when output is synthesized. MT on its own was 97% understandable. This differs from the finding reported in Somers & Sugita (2003), where Japanese ASR was quite reliable, and MT was the weak link. Chinese ASR is evidently considerably more difficult.

Taking ASR out of the equation still requires text to be input. Exploring the scenario of LEP patients wishing to read prescription labels, Ghobadi (2007) first experimented with a handheld






```
~R Cai.Dms 0.1s
put ~dr"~ into the nostrils
tWiCE jalnis directedr'
THROB QNAY OGYS N~IS-
OPENLY NOT aII@.
```

```
is. PREDN~SOL@E Teaks i~'r A1 of 3)
t~e1iio, -be taken Baily, reducing ile Diane
;er seven days as dire~liied bj yc'ur ~ucIjI-S
li THIS ~E ~~ AFTER
e. Fa_lo* Tie )~.~~IEU INiTRU~TIO~i
~"~ 'ijIj'~i%::"
```

Figure 1. Images of typical prescription labels and results of scanning with a handheld scanner.

scanner. If one considers that a typical prescription label is printed with a low-quality printer onto a label that is then often wrapped round a container, it is no surprise that scan results leave a lot of work still to be done (cf. Figure 1).

If instructions cannot be scanned in, we must devise some other text input method suitable for a user who does not know English, may not be familiar with the Roman script, and may be illiterate, even in their own language. The obvious solution is to have the labels translated at source, i.e. by the pharmacist, though this involves huge problems related to the pharmacist's legal obligation to verify the instructions on the label, which obviously they cannot do if they are written in a foreign language. Despite some political opposition, LEP is recognized in the US as a potential source of discrimination, and a 1998 Office of Civil Rights memorandum (OCR 1998) puts in place requirements for translations to be made available as part of healthcare provision. There is some evidence of use of MT (e.g. Sharif et al. 2006, Barclay 2007) where available, which of course always needs to be checked for translation accuracy, but this is not a viable solution for many of the languages needed. And even where the foreign language in question (Spanish) is well resourced, there is a reluctance to do so (Barclay 2007).

3 Spoken Language Translation vs. low-level technology

The problem of LEP patients has had some attention from the Language Technology community: so far, the focus has been on medical SLT systems, as mentioned above. We have elsewhere (Somers 2006) made some critical remarks about the direction some of this research has taken, and these are worth briefly repeating here in connection with our proposal that SLT – especially as currently implemented – is not always the most appropriate technology for all LEP patients' (and their clinicians') needs.

We have already mentioned the fact that current SLT systems inevitably see the doctor as being in control of the system and hence of the dialogue itself. Several assumptions underlying this set-up are false: the doctor's familiarity with computers in general and the SLT device in particular is assumed to be superior to the patient's (e.g. Narayanan et al. 2004:101, Precoda et al. 2004:9, Bouillon et al. 2007:42), but this may not be true, especially if the patient becomes a regular user. In our own research, admittedly with a much simpler device (Johnson 2007, Somers & Lovel 2007, Somers et al. in prep.), we found many patients more than willing to share or even take over control of the device, as shown in Figure 2a, in contrast to the scenario presented in the on-line video demo of an SLT system (Figure 2b), where the doctor (the one in the white coat) has total control to the extent that the patient is not even allowed to see the screen.

Sharing the device will also facilitate its use in promoting communication via a combination of technologies. Text and (where literacy is a problem) pictures can support the spoken (translated) word and even to a certain extent supplant speech: certain parts of doctor–patient

(a) Clinician and patient sharing the laptop device (from Somers & Lovel 2007)

(b) Snapshot from Transonics' demo movie (source: http://sail.usc.edu/transonics/demo/transedit02lr.mov, accessed 14 May 2007)

Figure 2. Contrasting perspectives in use of computer-based communication device by clinician and patient

dialogues (and indeed other exchanges on the pathway to healthcare) follow a fairly predictable pattern that can be exploited by using predetermined questions and (sets of) possible answers which, as we have discovered (Johnson 2007, Somers & Lovel 2007, Somers et al. in prep.) can lead to very high satisfaction rates, even though some drawbacks are recognized

In our research, in which as a test case we focused on Somalis with asthma-related conditions, we developed software on an ordinary laptop using a mousepad, and on a touch-screen tablet using a stylus, which permitted clinicians to choose freely from a range of 69 questions grouped under various topics. The questions were presented in both English and Somali, with pre-recorded (human) speech for both the questions and the possible answers on a screen as illustrated in Figure 3. The patient could review all the possible answers by clicking on the symbols before indicating to the clinician the desired answer.

We tested the software in simulated consultations with six GPs and asthma nurses and 26 Somali patients. All 26 simulations were completed adequately: none were abandoned due to difficulties using the system, with communication, or due to frustration on the part of Somalis or clinicians. In 20 of the 26 simulations, all questions were answered by the patients. Post-session feedback questionnaires indicated extremely high satisfaction ratings by both clinicians and patients with almost every aspect of the system (see Table 1): the only serious drawback noted was the rather obvious problem that the system did not allow the users to go off-script, as reflected in low clinician satisfaction scores for eliciting the patient's worries (42%) and building a relationship (69%), both key contributors to the overall goal of achieving a clinical outcome (65%).

Of course the system described does not involve MT in any sense. The reason for mentioning it here is to make the point that for some aspects of doctor–patient communication, where the content of the dialogue is sufficiently predictable, it might be safer to use a simpler technology such as that described here. We will surely need SLT for some communication tasks, but it makes sense, especially when the effort

Figure 3. Screen showing possible answers to the question "What kind of animal did you own in Somalia?". The question itself, and each of the possible answers is associated with a digitised recording in Somali.

	VS	S	other	%
Size of symbols (P)	25	1	0	100
Size of symbols (C) N=9	5	4	0	100
Size of text (P)	23	1	2	92
Size of text (C) N=9	4	5	0	100
Range of questions (P)	25	1	0	100
Range of questions (C) N=9	1	7	1	89
Range of responses (P)	21	3	2	92
Range of responses (C) N=9	3	5	1	89
Using laptop (P) N=14	3	3	8*	43
Using tablet (P) N=12	7	4	1	91
Using mousepad (P) N=14	3	3	8*	43
Using stylus (P) N=12	9	3	0	100
Navigation (P)	11	9	6*	77
Navigation (C)	14	12	0	100
P's ability to use device (P)	8	12	6*	77
P's ability to use device (C)	12	9	5*	81
C's ability to use device (P)	26	0	0	100
C's ability to use device (C)	9	15	0	100
P understand C's questions	23	3	0	100
C understand P's responses	10	13	3	88
P answer C's questions	22	4	0	100
C elicit information	12	11	3	88
Make self understood (P)	22	4	0	100
Make self understood (C)	8	15	3	88
P explain worries to C	11	4	1	96
C elicit P's worries	7	4	13	42
Build a relationship (P)	22	1	3	88
Build a relationship (C)	7	11	8	69
Better than no interp. (P)	22	3	1	96
Better than no interp. (C)	14	8	4	85
P satisfied with review	25	1	0	100
C achieved desired outcome	11	6	9	65

Table 1. Satisfaction ratings for a variety of questions. Key: "P" patient (N=26), "C" clinician (N=9). "VS" very satisfied, "S" satisfied, "Other" includes dissatisfied, very dissatisfied, don't know and (especially where marked *) not applicable. Except where indicated, N=26, corresponding to the number of sessions.

required to build SLT systems for certain languages is so great, to seek alternative solutions.

4 Conclusions

Spoken language translation and MT for under-resourced languages are two of greatest new challenges for the MT community. Putting them together gives a task that is almost impossible to contemplate at the present time. In this paper we have looked at one particular domain where the need for such technology is particularly important, and, in the spirit of the title of the TMI conference series, have put forward some theoretical and methodological issues related to that task. The main theoretical point made has been the need to focus on user-centered rather than technology-centered design in SLT. And regarding methodology, the point has been made that some lesser technologies, as well as some "cheats", may be the way forward, at least in the short term.

References

Bachman JW (2003) The patient-computer interview: a neglected tool that can aid the clinician. *Mayo Clinic Proceedings* 78, 67–78.

Barclay L (2007) Pharmacies may not always translate prescription labels for non-English speaking patients. [Report on presentation by L Weiss at Society for General Internal Medicine 2007 Annual Meeting, Toronto, Ont. Abstract 172022]. *Medscape Medical News*, April 27, 2007. Available at: http://www.medscape.com/viewarticle/555840. Accessed 14 May 2007.

Besacier L, Le V-B, Boitet C, Berment V (2006) ASR and translation for under-resourced languages. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, pp. V-1221–4.

Bischoff A, Bovier PA, Isah R, Francoise G, Ariel E, Louis L (2003) Language barriers between nurses and asylum seekers: their impact on symptom reporting and referral. *Social Science in Medicine* 57, 503–12.

Black AW, Brown RD, Frederking RF, Singh R, Moody J, Steinbrecher E (2002) TONGUES: Rapid development of a speech-to-speech translation system. *Proceedings of HLT 2002: Second International Conference on Human Language Technology Research*, San Diego, California, pp. 183–6.

Bouillon P, Flores G, Starlander M, Chatzichrisafis N, Santaholma M, Tsourakis N, Rayner M, Hockey BA (2007) A bidirectional grammar-based medical speech translator. *Proceedings of the Workshop on Grammar-based approaches to spoken language processing*, Prague, Czech Republic, pp. 41–48.

Bouillon P, Rayner M, Chatzichrisafis N, Hockey BA, Santaholma M, Starlander M, Nakao Y, Kanzaki K, IsaharaH (2005) A generic multi-lingual open source platform for limited-domain medical speech translation. *Proceedings of the Tenth Conference on European Association of Machine Translation,*

"Practical applications of machine translation", Budapest, Hungary, pp. 50–8.

Ehsani F, Kimzey J, Master D, Park H, Sudre K (2006) Rapid development of a speech translation system for Korean. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, pp. V-1225–8.

Ettelaie E, Gandhe S, Georgiou P, Belvin R, Knight K, Marcu D, Narayanan S, Traum D (2005) Transonics: A practical speech-to-speech translator for English-Farsi medical dialogues. *43rd Annual Meeting of the Association for Computational Linguistics: ACL-05 Interactive Poster and Demonstration Sessions*, Ann Arbor, MI, pp. 89–92.

Evans G, Polyzoaki K, Blenkhorn P (2002) An approach to producing new languages for talking applications for use by blind people. In K Miesenberger, J Klaus, W Zagler (eds) *8th ICCHP, Computers Helping People with Special Needs*, (LNCS 2398), Berlin: Springer Verlag, pp. 575–82.

Fassil Y (2002) Looking after the health of refugees. *British Medical Journal* 321, 59.

Flores G, Abreu M, Tomany-Korman SC (2005) Limited English proficiency, primary language at home, and disparities in children's health care: How language barriers are measured matters. *Public Health Report* 120, 418–30.

Ghobadi B (2007) Farsi prescription labels translator. Dissertation, School of Informatics, University of Manchester.

Jacobs EA, Lauderdale DS, Meltzer D, Shorey JM, Levison W, Thisted R (2001) Impact of interpreter services on delivery of health care to limited-English-proficient patients. *Journal of General Internal Medicine* 16, 468–74.

Johnson MJ (2007) An exploration into support for communication between health care practitioners and Somalis using assistive language technology in the context of asthma consultations. PhD thesis, School of Nursing, Midwifery and Social Work/School of Informatics, University of Manchester.

Jones D, Gill P (1998) Refugees and primary care: tackling the inequalities. *British Medical Journal* 317, 1444–6.

Kathol A, Precoda K, Vergyri D, Wang W, Riehemann S (2005) Speech translation for low-resource languages: The case of Pashto. *Interspeech'2005 – Eurospeech*, Lisbon, pp. 2273–6.

Narayanan S., Ananthakrishnan S, Belvin R, Ettelaie E, Gandhe S, Ganjavi S, Georgiou PG, Hein CM, Kadambe S, Knight K, Marcu D, Neely HE, Srinivasamurthy N, Traum, D, Wang D (2004) The Transonics spoken dialogue translator: An aid for English-Persian doctor-patient interviews. In T Bickmore (ed.) *Dialogue Systems for Health Communication: Papers from the 2004 Fall Symposium*, Menlo Park, California: American Association for Artificial Intelligence, pp. 97–103.

Narayanan SS, Georgiou PG, Sethy A, Wang D, Bulut M, Sundaram S, Ettalaie E, Ananthakrishnan S, Franco H, Precoda K, Vergyri D, Zheng J, Wang W, Gadde RR, Graciarena M, Abrash V, Frandsen M, Richey C (2006) Speech recognition engineering issues in speech to speech translation system design for low resource languages and domains. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, pp. V-1209–12.

OCR (Office of Civil Rights) (1998) Guidance memorandum: Title VI prohibition against national origin discrimination—persons with limited-English proficiency. January 29, 1998. Available at: http://www.hhs.gov/ocr/lepfinal.htm. Accessed 14 May 2007.

Orlovsky C (2005) Talking pill bottles let medications speak for themselves'. *NurseZone.com* (online magazine). Available at: www.nursezone.com/Job/DevicesandTechnology.asp?articleID=14396. Accessed 14 May 2007.

Osman L, Abdalla M, Beattie J, Ross S, Russell I, Friend J, Legge J, Douglas J (1994) Reducing hospital admissions through computer supported education for asthma patients. *British Medical Journal* 308, 568–71.

Precoda K, Franco H, Dost A, Frandsen M, Fry J, Kathol A, Richey C, Riehemann S, Vergyri D, Zheng J, Culy C (2004) Limited-domain speech-to-speech translation between English and Pashto. *HLT-NAACL 2004, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, Boston, MA, pp. 9–12.

Rizvi SM (in prep.) Faking Urdu speech recognition for a doctor–patient dialogue system. MSc dissertation, School of Informatics, University of Manchester.

Schultz T, Alexander D, Black AW, Peterson K, Suebvisai S, Waibel A (2004) A Thai speech translation system for medical dialogs. *HLT-NAACL 2004, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, Boston, MA, pp. 34–5.

Schultz T, Black AW (2006) Challenges with rapid adaptation of speech translation systems to new language pairs. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, pp. V-1213–16.

Sharif I, Lo S, Ozuah PO (2006) Availability of Spanish prescription labels. *Journal of Health Care for the Poor and Underserved* 17, 65–9.

Somers H (2006) Language Engineering and the pathway to healthcare: A user-oriented view. *HLT/NAACL-06 Medical Speech Translation, Proceedings of the Workshop*, New York, pp. 32–9.

Somers HL, Caress A-L, Evans DG, Johnson MJ, Lovel HJ, Mohamed Z (in prep.) A computer-based aid for communication between patients with limited English and their clinicians, using symbols and digitised speech. Submitted to *International Journal of Medical Informatics*.

Somers H, Evans G, Mohamed Z (2006) Developing speech synthesis for under-resourced languages by "faking it": An experiment with Somali. *5th International Conference on Language Resources and Evaluation*, Genoa, Italy, pp. 2578–81.

Somers H, Lovel H (2003) Computer-based support for patients with limited English. *Association for Computational Linguistics EACL 2003, 10th Conference of the European Chapter, Proceedings of the 7th International EAMT Workshop on MT and other language technology tools*, Budapest, pp. 41–9.

Somers HL, Lovel HJ (2007) Can AAC technology facilitate communication for patients with limited English? ESRC Project Final Report, School of Informatics, University of Manchester. Available at http://www.informatics.manchester.ac.uk/~harold/ESRCfinal.pdf. Accessed 8 August 2007.

Somers H, Sugita Y (2003) Evaluating commercial spoken language translation software. *MT Summit IX: Proceedings of the Ninth Machine Translation Summit*, New Orleans, pp. 370–7.

Stewart M, Brown JB, Weston WW, McWhinney IR, McWilliam CL, Freeman TR (2003) *Patient-centered medicine: Transforming the clinical method* (2nd ed.). Abingdon, Oxon: Radcliffe.

Wang B (2007) Chinese to English speech translation system built from standard components. Dissertation, School of Informatics, University of Manchester.

Westberg SM, Sorensen TD (2005) Pharmacy-related health disparities experienced by non-English-speaking patients: Impact of pharmaceutical care. *Journal of the American Pharmaceutical Association* 45, 48–54.

Zhou B, Déchelotte D, Gao Y (2004) Two-way speech-to-speech translation on handheld devices. *INTERSPEECH 2004 – ICSLP, 8th International Conference on Spoken Language Processing*, Jeju Island, Korea, pp. 1637–40.

Hand in Hand: Automatic Sign Language to English Translation

Daniel Stein, Philippe Dreuw, Hermann Ney
Computer Science Department,
RWTH Aachen University, Germany
{stein, dreuw, ney}
@i6.informatik.rwth-aachen.de

Sara Morrissey, Andy Way
School of Computing
Dublin City University, Ireland
{smorri,away}
@computing.dcu.ie

Abstract

In this paper, we describe the first data-driven automatic sign-language-to-speech translation system. While both sign language (SL) recognition and translation techniques exist, both use an intermediate notation system not directly intelligible for untrained users. We combine a SL recognizing framework with a state-of-the-art phrase-based machine translation (MT) system, using corpora of both American Sign Language and Irish Sign Language data. In a set of experiments we show the overall results and also illustrate the importance of including a vision-based knowledge source in the development of a complete SL translation system.

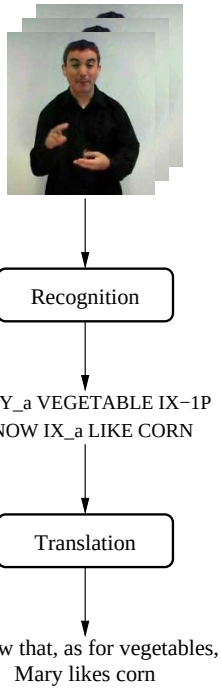


Figure 1: System setup with sample sentence

1 Introduction

The communication between deaf and hearing persons poses a much stronger problem than the communication between blind and seeing people. While the latter can talk freely by means of a common spoken language in which both are equally proficient, the deaf have their own, manual-visual language.

In this paper, we present an approach to automatically recognize sign language and translate it into a spoken language by means of data-driven methods. While the recognizer output is not easily intelligible because of different grammar and annotation format, we show that translation into the spoken language using standardized statistical machine translation (SMT) methods gives reasonable results, even for extremely small corpora. In preliminary experiments, we also give an outlook of how to incorporate vision-based features used in the recognizer to improve the overall translation result. Our work focuses on translating American Sign Language (ASL) and Irish Sign Language (ISL) into English (see Figure 1).

The remainder of the paper is constructed as follows. Section 2 introduces sign languages and gives an overview of the transcription methodology employed for capturing descriptions of sign

languages. The area of gesture recognition is presented in section 3. Section 4 details data-driven MT approaches for SLs and describes the MT system we have employed. The experiments carried out are described in section 5 and the results are discussed in section 6. Finally, we conclude the paper in section 7 and outline the future avenues for our work.

2 Sign Languages

In spite of common misconceptions, SLs are natural, indigenous and independent means of communication for deaf and hard-of-hearing communities worldwide. Since the languages have not been created artificially but rather evolved naturally, it is no surprise that most countries have their own particular SL as well as local dialects. SLs are grammatically distinct from spoken languages and the grammar makes extensive use of the possibilities of a visual/gestural modality: locations, verb inflections, pronouns and many other linguistic devices are conveyed by spatial information in front of the signer. Apart from the obvious employment of the hands as information carriers, SLs also use affected facial expressions, tilts of the head and shoulder as well as the velocity of the sign to incorporate information such as comparative degree or subclauses.

For example, ISL, one of the SLs used in this paper, is the primary language of the Irish Deaf community. Despite this, the language is not recognised as an official language in Ireland, however, the 5000 strong community is joined by the Irish Deaf Society[1] and the Centre for Deaf Studies[2] in promoting ISL awareness and research across the country.

2.1 Sign Language Transcription

One of the striking differences between signed and spoken languages is the lack of a formally adopted writing system for SLs. There have been some attempts to develop writing systems for SLs, many of which are based on the seminal work of (Stokoe, 1960) and describe the handshape, location and articulated movement of a sign. These include the Hamburg Notation System (HamNoSys) (Hanke, 2004) and SignWriting (Sutton, 1995). Developed as handwriting systems, they use simple line drawings that are intuitively and visually connected to the signs themselves.

Despite the development of these approaches, they currently fall short of being either computationally useful or comprehensive enough for use in SL MT. For this reason we have chosen to use an approach referred to as *annotation* (Pizzuto and Pietrandrea, 2001). This involves the manual transcription of sign language taken from video data that is reproduced in a *gloss* format. The gloss is a semantic representation of sign language where, conventionally, the semantic meaning of the sign is transcribed in the upper case stem form of the local spoken language. The annotation "IX" signifies a deictic reference signed by a pointing gesture with the index finger. Additional spatial and non-manual information may also be added. An example of annotated glosses taken from our data is shown in Table 1. The first sentence is written in ASL glosses. The narrator (indicated by IX-1P) knows that Mary, at the spatial position referenced as "_a" and in the subordinate clause, likes corn. Here, the deixis "IX_a" serves as a pronoun to pick up the object of the subordinate clause again. A second sentence closer to the English grammar is written in ISL glosses. Note that, although both ISL and ASL are glossed in English, the grammar and vocabularies of the two sign languages are completely different.

2.2 The Corpus

Data-driven approaches to MT require a bilingual data set. In comparison to spoken language translation, SL corpora are difficult to acquire. To tune and test our system, we assembled the RWTH-Boston-104 corpus as a subset of a larger database of sign language sentences that were recorded at Boston University for linguistic research (Neidle et al., 1999). The RWTH-Boston-104 corpus consists of 201 video sentences, consisting of 104 unique words. The sentences were signed by 3 speakers and the corpus is split into 161 training and 40 test sequences. An overview of the corpus is given in Table 2: 26% of the training data are singletons, i.e. we only have one attempt to train the models properly. The sentences

[1] http://www.deaf.ie

[2] http://www.centrefordeafstudies.com

Table 1: Gloss annotation examples

ASL gloss	MARY_a VEGETABLE IX-1P KNOW IX_a LIKE CORN
English translation	I know that, as for vegetables, Mary likes corn.
ISL gloss	IX-FLIGHT FLIGHT B A ROUND TRIP IX-FLIGHT palm-up
English translation	Is flight B A a round trip flight?

Table 2: RWTH-Boston-104 corpus statistics

	Training	Test
sentences	161	40
running words	710	178
unique words	103	65
singletons	27	9
OOV	-	1

Table 3: ATIS corpus statistics

	Training	Devel	Test
sentences	482	98	100
running words	3707	593	432
unique words	375	88	128
singletons	144	28	10
OOV	-	30	4

have a rather simple structure and therefore the language model perplexity is low. The test corpus has one out-of-vocabulary (OOV) word. Obviously, this word cannot be recognized correctly using whole-word models.

Apart from this relatively small corpus, few data collections exist that are interesting for data-driven approaches. Much of what is available is in the form of conversation, stories and poetry which is unsuitable for ASLR and MT as illustrated in (Morrissey and Way, 2006). For this reason we chose to create our own corpus. We used the Air Travel Information System (ATIS) corpus of transcriptions from speech containing flight information in English as our base. The corpus consists of 680 sentences. For the purposes of our translation work, we had the data set translated and signed into ISL by native deaf signers. This was then manually annotated with semantic glosses as described in section 2.1.

3 Sign Language Recognition

The automatic sign language recognition (ASLR) system is based on an automatic speech recognition (ASR) system adapted to visual features(Lööf et al., 2006). The word sequence which best fits the current observation to the trained word model inventory (which is related to the acoustic model in ASR) and language model (LM) will be the recognition result.

In our baseline system, we use intensity images scaled to 32×32 pixels as features. To model image variability, various approaches are known and have been applied to gesture recognition similar to the works of (Dreuw et al., 2007). The baseline system is Viterbi trained and uses a trigram LM. In subsequent steps, this baseline system is extended by features that take the hand position and movement into account.

To extract manual features, the dominant hand is tracked in each image sequence. Therefore, a robust tracking algorithm is required as the signing hand frequently moves in front of the face, may temporarily disappear, or cross the other hand. We use an approach based on dynamic programming which is inspired by the time alignment algorithm in ASR and which is guaranteed to find the optimal path with respect to a given criterion and which prevents taking possibly wrong local decisions. Given the position of the hand, features such as velocity, trajectory, and acceleration can easily be extracted.

4 Data-driven Sign Language MT

SL MT is still a new area of research with work dating back only roughly a decade. Despite the relative novelty of the area in comparison with mainstream MT, it has followed the trend away from 'second generation' rule-based approaches towards data-driven methods. An overview of current developments in this area is given in section 4.1 and the translation system used for our experiments is described in section 4.2.

4.1 Related Research

There are currently four groups working on data-driven SL MT. Their approaches are described below:

- (Morrissey and Way, 2005) have explored Example-Based MT approaches for the language pair English–Sign Language of the Netherlands with further developments being made in the area of ISL.
- (Stein et al., 2006) have developed an SMT system for German and German sign language in the domain weather reports. Their work describes the addition of pre- and post-processing steps to improve the translation for this language pairing. However, the methods rely on external knowledge sources such as grammar parsers that cannot be utilized here since our source input are glosses, for which no automatic parser exists.
- (Chiu et al., 2007) present a system for the language pair Chinese and Taiwanese sign language. The optimizing methodologies are shown to outperform IBM model 2.
- (San-Segundo et al., 2006) have undertaken some basic research on Spanish and Spanish sign language with a focus on a speech-to-gesture architecture. They propose a decompensation of the translation process into two steps: first they translate from written text into a semantic representation of the signs. Afterwards a second translation into graphically oriented representation is done. This representation can be understood by the avatar. Note, however, that this is the opposite translation direction as the one proposed here.

4.2 Statistical Machine Translation

We use a state-of-the-art phrase-based statistical machine translation system to automatically transfer the meaning of a source language sentence into a target language sentence.

Following the notation convention, we denote the source language with J words as $f_1^J = f_1 \dots f_J$, a target language sentence as $e_1^I = e_1 \dots e_I$ and their correspondence as the *a posteriori* probability $\Pr(e_1^I|f_1^J)$. Our baseline system maximizes the translation probability directly using a log-linear model:

$$p(e_1^I|f_1^J) = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum_{\tilde{e}_1^I} \exp\left(\sum_{m=1}^M \lambda_m h_m(\tilde{e}_1^I, f_1^J)\right)}$$

with a set of different features h_m, scaling factors λ_m and the denominator a normalization factor that can be ignored in the maximization process. We choose the λ_m by optimizing an MT performance measure on a development corpus using the downhill simplex algorithm.

For a complete description of the system, see (Mauser et al., 2006).

5 Experiments

5.1 RWTH-Boston-104

Baseline. The baseline translation of the annotated gloss data into written English for the RWTH-Boston-104 has a word error rate (WER) of 21.2% and a position-independent word error rate (PER) of 20.1%. Looking at the data, the translation is even more accurate than that – the main problem being the lack of sentence boundary markers like dots and commas in sign language which are then omitted in the translation process.

Recognition. First, we analyze different appearance-based features for our baseline system. The simplest feature is to use intensity images down scaled to 32×32 pixels. As a baseline, we obtained a WER of 33.7%. For reducing the number of features and thus the number of parameters to be learned in the models, we apply linear feature reduction technique to the data, the principal component analysis (PCA). With PCA, a WER of 27.5% can be obtained (see Figure 2).

A log-linear combination of two independently trained models (PCA that include automatic tracking of hand velocity (HV) and hand trajectory (HT), respectively), leads to our best result of 17.9% WER (i.e. 17 del., 3 ins., and 12 subst.), where the model weights have been optimized empirically.

Sign-Language-to-Speech. If we translate these recognized glosses into written English (again, with punctuation mark post-processing), the overall score is 27.6% WER and 23.6% PER.

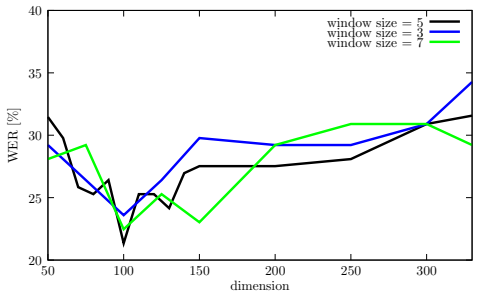


Figure 2: Combination of PCA-frames using PCA windowing

Figure 3: Sample frames for pointing near and far used in the translation.

In another set of experiments, we derive the tracking positions from all of the sentences. The positions of both hands have been annotated manually for 1119 frames in 15 videos. We achieve a 2.30% tracking error rate for a 20×20 search window (Dreuw et al., 2006). In order to distinguish between locative and descriptive pronouns, the tracking positions of the dominant-hand were clustered and their mean calculated. Then, for deictic signs, the nearest cluster according to the Euclidean distance was added as additional word information for the translation model (see Figure 3).

In the translation, the incorporation of the tracking data for the deixis words helped the translation system to discriminate between deixis as distinctive article, locative or discourse entity reference function. For example, the sentence "`JOHN GIVE WOMAN IX COAT`" might be translated into "*John gives the woman the coat*" or "*John gives the woman over there the coat*" depending on the nature of the pointing gesture "`IX`". Using the tracking data, the translation improves in performance from 28.5% WER to 26.5% and from 23.8% PER to 23.5%.

5.2 ATIS Corpus

The baseline translation of the annotated gloss data into written English has a WER of 45.1% and a PER of 34.7%. While this is a much more challenging result in itself if introduced with an additional error source like recognition, the preliminary recogniton of the ATIS videos had an error rate of 85% WER, with 327 deletions, 5 insertions and 175 substitutions out of 593 words. It is apparent from these result that further translation makes no sense at the moment if we start from the recognized data.

6 Discussion

Although the size of the corpus RWTH-Boston-104 is far too small to make reliable assumptions about the general significance of the results, at the very least we show that statistical machine translation is capable to work as an intermediate step for a complete sign-to-speech system. Even for extremely small training data, the resulting translation quality is reasonable.

We have shown that the recognition output in itself is not directly intelligble, given the different grammar and vocabulary of sign languages and shortages of the existing annotation system, but together with the automatic translation, the overall system can be easily trained on new language pairs and new domains. This set of sentences could without any doubt be translated with a reasonable rule-based system, yet it is not the ultimate goal to translate this corpus but to show that a sign-to-speech system is in principle possible using statistical methods, given reasonable data.

Moreover, adding features from the recognition process like the hand tracking position seems to help the translation quality, as it enables the system to distinguish between certain flexions of common words like the pointing gesture "`IX`". We argue that this can be compared to adding parts-of-speech (POS) information, to discriminate for example between deixis as distinctive article or as locative discourse entity reference.

As no grammar parser exists for sign language annotation, we propose a stemming of the glosses (i.e. leaving out the flexion) during recognition to cope with data sparseness problems. The missing

information can be included by adding the relevant features during the translation process, analogous to morpho-syntactic translation enhancement to sparse language pairs with a rich grammatical parser on the source language side.

For the more sophisticated ATIS Corpus, translation is possible, at this stage, however, recognition produces far too much noise for a reasonable translation adaption. Given the numbers of singletons alone, these are already quite an obstacle for translation, but if they consist of several frames in a video where the exact starting and end time is not passed on to the recogniser, they are quite challenging for the algorithm. Moreover, sign languages produce quite a large effect known as coarticulation, i.e. the movement between two regular signs, that cannot be as easily trained. To date, we have not carried out experiments on the ATIS data with the addition of several recognition features, so, while time-expensive, there is still ground for improved results. The ratio of the deletions with regard to the number of words also strongly indicate that there is much room for improvement with tuning on the development set.

7 Conclusion

To the best of our knowledge, we present the first approach to combine data-driven methods for recognition output and translation of sign languages. Both these methods alone work on an intermediate notation, that does not provide any support for the target group as it is not used in the deaf community. With our system, we are able to produce a unique sign-language-to-speech system.

Like other poorly resourced languages, sign language research suffers from lack of training material to feed the corpus-based algorithms properly. However, given the data sparseness, a small domain that matches in vocabulary size according to the small sentence number, gives reasonably optimistic results.

We have also shown that the translation improves if it relies on additional recognition data and argue that this can be interpreted as adding external POS information. Other features are likely to improve the error rates as well and should be investigated further, these include: velocity movements, head tracking to measure the tilt of the head (often indicating sub-clauses) or the shift of the upper body (possible indications for direct or indirect speech). Also, a complex entity model can be built up based on the location of the signs. If a new character in the discourse is introduced and stored on the right hand-side of the chest, later deictic pronoun signs pointing to the same position can be interpreted correctly, while pronouns in spoken languages are usually ambiguous.

References

[Chiu et al.2007] Y.-H. Chiu, C.-H. Wu, H.-Y. Su, and C.-J. Cheng. 2007. Joint Optimization of Word Alignment and Epenthesis Generation for Chinese to Taiwanese Sign Synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **29**(1):28–39.

[Dreuw et al.2006] P. Dreuw, T. Deselaers, D. Rybach, D. Keysers, and H. Ney. 2006. Tracking using dynamic programming for appearance-based sign language recognition. In *7th Intl. Conference on Automatic Face and Gesture Recognition*, IEEE, pages 293–298, Southampton, April.

[Dreuw et al.2007] Philippe Dreuw, David Rybach, Thomas Deselaers, Morteza Zahedi, and Hermann Ney. 2007. Speech recognition techniques for a sign language recognition system. In *Interspeech 2007 - Eurospeech*, page accepted for publication, Antwerp, Belgium, August.

[Hanke2004] T. Hanke. 2004. HamNoSys - Representing Sign Language Data in Language Resources and Language Processing Contexts. In *Workshop on the Representation and Processing of Sign Languages at LREC 04*, pages 1–6, Lisbon, Portugal.

[Lööf et al.2006] J. Lööf, M. Bisani, C. Gollan, G. Heigold, B. Hoffmeister, C. Plahl, R. Schluter, and H. Ney. 2006. The 2006 RWTH parliamentary speeches transcription system. In *Ninth ICSLP*, Pittsburgh, Pennsylvania, September.

[Mauser et al.2006] A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney. 2006. The RWTH statistical machine translation system for the IWSLT 2006 evaluation. In *IWSLT*, pages 103–110, Kyoto, Japan, November. Best paper award.

[Morrissey and Way2005] S. Morrissey and A. Way. 2005. An Example-based Approach to Translating Sign Language. In *Proceedings of the Workshop in Example-Based Machine Translation (MT Summit X)*, pages 109–116, Phuket, Thailand.

[Morrissey and Way2006] S. Morrissey and A. Way. 2006. Lost in Translation: the Problems of Using

Mainstream MT Evaluation Metrics for Sign Language Translation. In *Proceedings of the 5th SALTMIL Workshop on Minority Languages at LREC 2006*, pages 91–98, Genoa, Italy.

[Neidle et al.1999] C. Neidle, J. Kegl, D. MacLaughlin, B. Bahan, and R.G. Lee. 1999. *The Syntax of American Sign Language*. MIT Press.

[Pizzuto and Pietrandrea2001] E. Pizzuto and P. Pietrandrea. 2001. The notation of signed texts: open questions and indications for further research. *Sign Language and Linguistics (Special Issue - Sign Transcription and Database Storage of Sign Information)*, 4: 1/2:29–43.

[San-Segundo et al.2006] R. San-Segundo, R. Barra, L. F. D'Haro, J. M. Montero, R. Córdoba, and J. Ferreiros. 2006. A Spanish Speech to Sign Language Translation System for assisting deaf-mute people. In *Proceedings of Interspeech 2006*, Pittsburgh, PA.

[Stein et al.2006] D. Stein, J. Bungeroth, and H. Ney. 2006. Morpho-Syntax Based Statistical Methods for Sign Language Translation. In *Proceedings of the 11th Annual conference of the European Association for Machine Translation (EAMT'06)*, pages 169–177, Oslo, Norway.

[Stokoe1960] W. C. Stokoe. 1960. *Sign language structure: an outline of the visual communication system of the American deaf*. Studies in Linguistics, Occasional Paper, 2nd printing 1993: Burtonsville, MD: Linstok Press.

[Sutton1995] V. Sutton. 1995. *Lessons in Sign Writing, Textbook and Workbook (Second Edition)*. The Center for Sutton Movement Writing, Inc.

A Cluster-Based Representation for Multi-System MT Evaluation

Nicolas Stroppa, Karolina Owczarzak
School of Computing
Dublin City University
Glasnevin, Dublin 9, Ireland
{nstroppa,owczarzak}@computing.dcu.ie

Abstract

Automatic evaluation metrics are often used to compare the quality of different systems. However, a small difference between the scores of two systems does not necessary reflect a real difference between their performance. Because such a difference can be significant or only due to chance, it is inadvisable to use a hard ranking to represent the evaluation of multiple systems.

In this paper, we propose a cluster-based representation for quality ranking of Machine Translation systems. A comparison of rankings produced by clustering based on automatic MT evaluation metrics with those based on human judgements shows that such interpretation of automatic metric scores provides dependable means of ordering MT systems with respect to their quality. We report experimental results comparing clusterings produced by BLEU, NIST, METEOR, and GTM with those derived from human judgement (of adequacy and fluency) on the IWSLT-2006 evaluation campaign data.

1 Introduction

Automatic evaluation metrics for Machine Translation (MT) have been given a lot of attention in the recent years, as their importance for MT research is hard to ignore. They are extremely useful in comparisons of developmental stages of an MT system, helping to test the influence of various parameters on the final translation output: addition or modification of rules in rule-based MT systems, modification of training settings for data-driven MT systems, etc. Moreover, they are also often used to compare the quality of different systems. Several evaluation campaigns strongly rely on automatic evaluation metrics (NIST, 2006; Paul, 2006) as well as on human judgment, which remains the ultimate evaluation schema, to assess the quality of participating MT systems.

The rankings of MT systems obtained with automatic evaluation metrics or human judgment are not strict in the sense that those scores may not be sufficient to distinguish between two systems. Indeed, a small difference between two scores does not necessary reflect a real difference between the performance of two systems. To test if the difference between the scores of two systems is *significant* or only due to chance, we can employ statistical significance tests using bootstrap (Efron and Tibshirani, 1993; Koehn, 2004) or approximate randomization (Noreen, 1989; Riezler and Maxwell, 2005) methods. This enables us to introduce a cluster-based representation which we feel is better suited to the ranking of system scores than a strict ranking which might be based on insignificant or accidental differences.

The quality of an automatic metric is often assessed by computing its correlation with human judgment (of adequacy and fluency) on a segment or system level. For an automatic evaluation metric, a high correlation with human judgment denotes a capability to correctly identify the quality of an MT system. In this paper, instead of com-

puting the direct correlation between automatic scores and human scores on a segment level, or in a hard ranking on a system level, we compare the clusters produced by automatic metrics and human judgements using an adaptation of the Rand statistic. In other terms, in this context, a metric will be considered good if it ranks various systems in the same order and groups them in the same clusters as human evaluators. We extend our analysis to clusterings produced by several automatic MT evaluation metrics: BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), and GTM (Melamed et al., 2004), using the evaluation data from the IWSLT-2006 evaluation campaign (Paul, 2006).

The remainder of this paper is organized as follows. In Section 2, we introduce the automatic evaluation metrics we tested. In Section 3, we present a statistical significance test based on approximate randomization, the cluster-based representation for ranking, and the cluster comparison strategies. In Section 4, we report experimental results. Section 5 concludes the paper and gives avenues for future work.

2 Automatic Evaluation Metrics

Since the introduction of BLEU (Papineni et al., 2002), a large number of other metrics have been developed, but the string-based metrics like BLEU, NIST (Doddington, 2002), GTM (Melamed et al., 2004), and METEOR (Banerjee and Lavie, 2005) have remained among the most popular, therefore we focus our analysis on them.

2.1 BLEU

The most popular evaluation metric BLEU (BiLingual Evaluation Understudy, (Papineni et al., 2002)) is based on a simple calculation of *modified precision*. Modified precision counts the number of n-grams in the translation that match at least one of the references and caps the count by the maximum number of occurrences of a given n-gram in a single reference. In other words, if a translation consists entirely of the word *the* repeated five times, but in one of the references *the* appears only once, and in the other only twice, we are allowed to count only two of the five matching words. This process is applied to any n, but in practice n-grams up to four are used. The modified precision results *for the whole document* at each n-gram level are combined together using geometric average. Moreover, in order to prevent unfair high precision scores for very short sentences, a brevity penalty is calculated over the test set, if the combined length of the translation segments is equal to or shorter than the combined length of best matching (closest in length) reference segments.

Note that BLEU was developed with document- or system-level evaluation in mind, and its construction does not allow for high correlation with human judgment on the level of individual segments. At segment level, many sentences will be scored as zero for not providing at least one four-gram in common with the references, which artificially levels down their quality. Segments shorter than four elements will be scored as zero irrespective of the number of lower n-gram matches. These effects are exacerbated as the number of available references decreases.

2.2 NIST

NIST was developed on the basis of BLEU-style n-gram calculation, but several improvements were added to raise the metric's correlations with human judgments (Doddington, 2002). Instead of geometric average, arithmetic average is used to combine results from all levels up to five grams, and the brevity penalty was adjusted to minimize the impact of small length variations. Most importantly, all n-grams are weighted according to their information with respect to the reference sentences, so that rarer and more informative sequences present in the translation will contribute more to the final score than sequences that are more common, and thus less informative.

2.3 GTM

Exploring a different avenue of research, GTM uses the standard notions of precision, recall, and their composite F-measure, to evaluate translation quality (Melamed et al., 2004). It calculates the word overlap between the translation and the reference(s), preventing double-counting when a word occurs multiple times, and it caps the resulting number of matches by the mean length of the references. While it also has the option of weight-

ing contiguous sequences more than unconnected matching fragments, Turian et al. (2003) conclude from their experiments that such a weight lowers the correlation with human judgment. In this work, we thus use the unweighted version of GTM. Turian et al. (2003) also show that GTM outperforms both BLEU and NIST with respect to correlation, irrespective of the number of references available.

2.4 METEOR

The evaluation in METEOR (Banerjee and Lavie, 2005) proceeds in several stages. First, all exact matches between the translation and the reference are found; next, the remaining words are stemmed and the matching process repeats; finally, there is the option of using WordNet to find matches between synonyms among the remaining non-matched words. The final score combines precision and heavily weighted recall at the unigram level with a penalty for non-contiguous matches.

3 Comparing Multiple Systems

3.1 Statistical Significance Testing using Approximate Randomization

Since a small difference between the scores of two systems does not necessary reflect a real difference between their performance, it is important to identify when this difference is *significant* or only due to chance. To discriminate between these two cases, we assume a null hypothesis which states that the two systems are of the same quality, and we consider the difference between their scores as significant only if we find statistical evidence indicating that the null hypothesis is false (with a certain degree of confidence).

When assumptions can be made about the probability distributions yielding the scores, it is possible to employ parametric methods such as the Student's t-test. When no specific assumption can be made, as it is the case for automatic evaluation metrics, we have to resort to non-parametric methods, such as bootstrap (Efron and Tibshirani, 1993; Koehn, 2004) or approximate randomization (Noreen, 1989; Riezler and Maxwell, 2005) methods. To use bootstrap, one would have to take the translation output of each MT system, produce a large number of samples from that output using sampling with replacement, and then create clusters of MT systems by collecting those with overlapping confidence intervals. However, in this paper we consider approximate randomization rather than bootstrap, following Riezler and Maxwell (2005) and Collins et al. (2005), who suggest that approximate randomization is more appropriate in such a context.

To compare the output of two systems using approximate randomization, we proceed as follows. First, we assume that we have access to n translations of the same sentences for the two systems. These translations are respectively denoted T (for system 1) and T' (for system 2), with $|T| = |T'| = n$. The set of reference translations for these sentences is denoted R. The score for T and T' are respectively $s = M(T, R)$ and $s' = M(T', R)$, where M denotes some metric (e.g. BLEU); their difference is $s - s'$.

Then, we build k new pairs of translation sets obtained by randomly permuting the translations in T and T', yielding the pairs $(T_1, T'_1), \ldots, (T_k, T'_k)$. For each $i \in 1..k$, the shuffle (T_i, T'_i) is obtained as follows: each pair of sentence in (T, T') is randomly shuffled with probability 0.5. Intuitively, if system 1 is better that system 2, then we obtain a lower score for the translations in T_i than for those in the original T, since T_i is obtained by replacing some translations in T with some translations from T' of lower quality. Consequently, in this scenario, we have $M(T_i, R) < M(T, R)$; similarly, we would also expect $M(T'_i, R) > M(T', R)$. In short, we expect the newly created T_i to be of lower quality than the original T.

$$M(T_i, R) - M(T'_i, R) < M(T, R) - M(T', R).$$

If this inequality is verified for $i \in 1..k$, we set $v_i = 0$, and $v_i = 1$ otherwise. If system 1 is better that system 2, then we expect $\sum_{i=1}^{k} v_i$ to be close to 0. On the contrary, if system 1 is not significantly better than system 2, then shuffling translations has little effect on the difference between the scores obtained, and $\sum_{i=1}^{k} v_i$ is unlikely to be close to 0. The p-value is simply computed as fol-

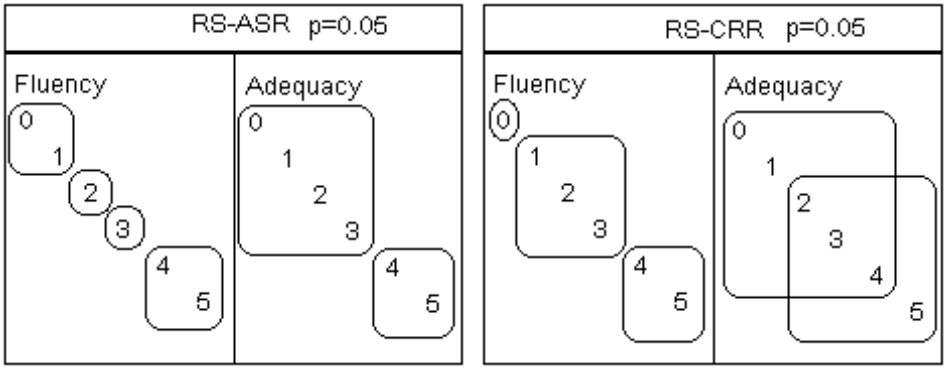


Figure 1: Examples of clusterings. Numbers 0-5 represent MT systems; clusters are created on the basis of fluency and adequacy scores. Relative height of the clusters shows their order.

lows:

$$p = \frac{(\sum_{i=1}^{k} v_i) + 1}{k}.$$

The null hypothesis is rejected if p is less than or equal to a specified rejection level, traditionally set to 0.05. In all our experiments, we used $k = 1000$ shuffles. We use the same method for all the considered metrics, including human judgement.

3.1.1 Implementation Issues

In order to compute statistical significance using approximate randomization, the values $M(T_i, R)$ and $M(T_i', R)$ are required for each shuffle (T_i, T_i'). However, even for document-level metrics such as BLEU, we do not have to compute BLEU for each shuffle. Indeed, it is sufficient to keep some information about each sentence (for BLEU: number of matching n-grams, lengths, etc.), and to aggregate them.

Consequently, the potentially expensive comparison between the reference sentences and the test sentences is performed once; only the aggregation of the sentence-level information, which is fast and cheap, is performed $k = 1000$ times. The computation of statistical significance for a test set of 500 sentences, with $k = 1000$ shuffles takes about 0.3 second for BLEU, and 0.7 second for NIST on a Pentium 4 processor, 3GHz.[1]

3.2 A Cluster-Based Representation

Most if not all comparisons of different MT sytems, including large-scale evaluations conducted in shared MT tasks, is done using a hard ranking of the participating sytems based on the system-level scores. However, as has been noted already, the difference in scores between two MT sytems may not be significant. We feel therefore that such strict rankings are inadvisable and not completely fair to the participating systems. In order to represent the ranking of MT systems according to their scores, we thus propose a cluster-based representation. In this representation, two systems are placed in the same cluster if they cannot be proven to differ in quality, i.e. if we have not succeeded in discarding the null hypothesis using approximate randomization. A cluster thus contains systems that are pairwise indistinguishable. By performing this comparison for all pairs of systems, this approach yields an ordered set of clusters. Formally, the method is expressed as follows. We note $s_1, s_2, \ldots, s_n$ the scores of n systems. We note $s_1 \gg s_2$ if s_1 is significantly higher than s_2, and $s_1 \sim s_2$ if their difference is not statistically significant. Using this cluster-based representation, we obtain an ordered set of clusters $C_1, \ldots, C_m$, such that:

$$\forall i \in 1..m,\ \forall k, l \in C_i,\ s_k \sim s_l,$$

$$\forall i, j \in 1..m,\ s.t.\ i < j,$$
$$\exists k \in C_i, l \in C_j, s_k \gg s_l.$$

This representation is suited to the ranking of system scores, and differs from the initial hard ranking, because one system can belong to several clusters. By using different p-values, we may

[1] Our C++ implementation, called FastMtEval, can be freely downloaded from `http://www.computing.dcu.ie/~nstroppa/softs/fast_mt_eval.tgz`.

obtain different cluster-based representations: the smaller the p-value, the bigger the clusters. An example of such a representation is given in Figure 1.

3.3 Comparing Clusters

In this section, we introduce a simple method to compare two clusterings. Our method is actually a simple adaptation of the Rand statistics (cf. Halkidi et al. (2001)), a method that can be used to compare non-ordered clusterings. The adaptation we propose aims at dealing with the ordered nature of the clusterings we consider.

A clustering C of n systems is a ordered set of clusters $C = \{C_1, \ldots, C_m\}$ such that $\forall i \in 1..m$, $C_i \subseteq 1..n$, and $\cup_i^m C_i = 1..n$. Let us recall that a system may belong to several clusters, i.e. we do not have necessarily $C_i \cap C_j = \emptyset$ for $i \neq j$.

To compare two clusterings C and D, we rely on a pairwise comparison of systems, i.e. clusterings C and D will be considered similar if for all pairs (i, j) of systems, C and D agree on the fact that systems i and j should be put on the same cluster or not. The Rand statistics counts the number of such agreements and divides it by the total number of comparisons, i.e. $\frac{n\times(n-1)}{2}$. In the ordered case, we have to add another factor. Indeed, if C and D agree that i and j should be placed on different clusters, but C says that i is significantly better than j and D shows the opposite, there is a strong disagreement between the clusterings. For a clustering C, we note $C(i, j)$ the relationship between the systems i and j according to the clustering. We have $C(i, j) \in \{'\sim', '\ll', '\gg'\}$. We have $'\sim'$, $'\ll'$, and $'\gg'$ respectively when i and j are indistinguishable, when j is significantly better than i, and when i is significantly better than j. The scoring is as follows:

$$s(c, d) = \begin{cases} 1 & \text{if } (c = d) \\ -1 & \text{if } (c = '\ll') \text{ and } (d = '\gg') \\ -1 & \text{if } (d = '\ll') \text{ and } (c = '\gg') \\ 0 & \text{otherwise.} \end{cases}$$

The first case corresponds to an agreement, the second and third cases are strong disagreements, and the last one is a weak disagreement. Our comparison metric is then computed as follows:

$$S(C, D) = \frac{2 \times \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} s(C(i, j), D(i, j))}{n \times (n - 1)},$$

which yields a value between -1 and 1. A value of -1 denotes a complete disagreement on the ranking, while a value of 1 denotes a complete agreement.

For example, the "similarity" between the two clusterings associated with fluency and adequacy on the left of Figure 1 is 0.67. Indeed, they agree on the following (10) pairs: $(0, 1)$, $(0, 4)$, $(0, 5)$, $(1, 4)$, $(1, 5)$, $(2, 4)$, $(2, 5)$, $(3, 4)$, $(3, 5)$, $(4, 5)$, and (weakly) disagree on the following pairs: $(0, 2)$, $(0, 3)$, $(1, 2)$, $(1, 3)$, $(2, 3)$, which gives a final score of $\frac{2\times10}{6\times5} = 0.67$.

4 Experimental Results

4.1 Data

The experiments were carried out using the Chinese–English datasets provided within the IWSLT 2006 evaluation campaign (Paul, 2006), extracted from the Basic Travel Expression Corpus (BTEC) (Takezawa et al., 2002). This multilingual speech corpus contains sentences similar to those that are usually found in phrase-books for tourists going abroad. Three input conditions are considered: continuous speech (CS-ASR), read-speech ASR (RS-ASR), and read-speech CRR (RS-CRR). In the first condition, the sentences to translate correspond to natural continuous speech; in the second case, the sentences are read and the input to translate comes from an ASR (Automatic Speech Recognition) system; in the last condition, MT systems are given the correct recognition results. For each conditions, 6 systems are considered. Since the various conditions corresponds to different views of the same sentences, it is possible to "merge" all the conditions together, in order to compare a total of 18 different systems (referred to as Mixed Track). The outputs of all systems were evaluated with respect to both adequacy and fluency. Automatic evaluation is performed using BLEU, NIST, METEOR, and GTM-1, with 7 references.

4.2 Cluster-Based Rankings

For each input condition and each metric, we constructed cluster-based rankings to represent the

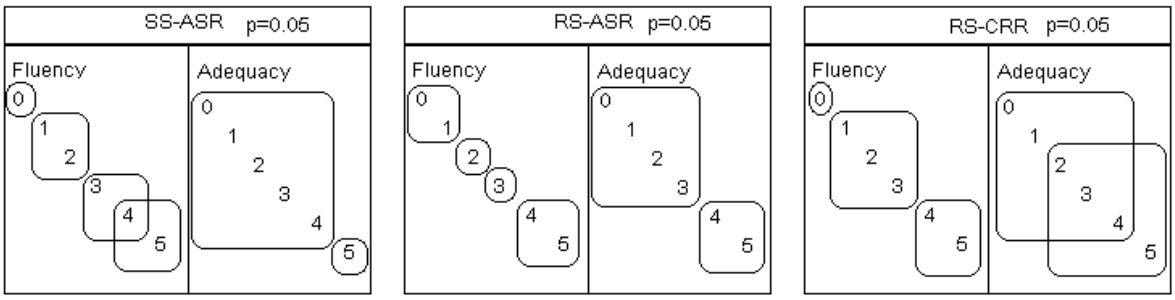


Figure 2: Clusterings of MT systems based on human judgements of fluency and adequacy

results obtained by the different systems. For those rankings, the level to test statistical significance is set to $p = 0.05$. The results for fluency and adequacy are displayed in Figure 2. Note that in this figure systems are numbered with respect to their rank according to a metric, i.e. system 0 in the fluency clustering is the best system according to fluency, and may be different from the system 0 in the adequacy clusterings.

We can observe that adequacy scores do not strongly differentiate the various participating systems, and the resulting clusters are big. In the case of fluency, there are more differences and systems are easier to distinguish. We also observe overlapping cases, in which a system belongs to several clusters.

To examine the influence of the significance level on the construction of the clusterings, we performed some tests with different values for p: 0.001, 0.002, 0.005, 0.01, 0.02, and 0.05. For the condition SS-ASR, we report the obtained results in Figure 3.

As expected, with a very high significance level ($p = 0.001$) it is not possible to distinguish between systems, and they are all placed in the same cluster, with respect to fluency as well as adequacy. Overall, however, the clusterings seem pretty stable: there are very few modifications between the clusterings with the p values 0.002, 0.005, 0.01, 0.02, and 0.05. For fluency, they are actually identical for the values 0.005, 0.01, and 0.02. For adequacy, they are identical for the values 0.002, 0.005, and 0.01. (See also Section 4.4 for a discussion about the choice of a significance level.)

4.3 Clusterings Comparison

Once constructed, we can compare the clusterings obtained with different evaluation metrics, using the comparison strategy introduced in Section 3.3 (with a p-value of 0.05). In particular, we computed the comparison scores between the automatic evaluation metrics BLEU, NIST, GTM-1, and METEOR, and the human judgement for fluency and adequacy. The results are displayed in Table 1.

		Fluency	Adequacy
SS-ASR	BLEU	0.47	0.4
	NIST	0	0.6
	METEOR	0	0.53
	GTM	-0.13	0.6
RS-ASR	BLEU	0.47	0.33
	NIST	0.4	0.27
	METEOR	0.33	0.13
	GTM	0.2	0.2
RS-CRR	BLEU	0.73	0.47
	NIST	0.4	0.27
	METEOR	0.53	0.26
	GTM	0.33	0.33
Mixed Track	BLEU	**0.58**	0.70
	NIST	0.34	0.64
	METEOR	0.39	**0.71**
	GTM	0.31	0.70

Table 1: Clustering comparison scores

According to these comparison scores, BLEU and METEOR seem to be better than NIST and GTM at finding rankings similar to those obtained with human judgement. In particular, BLEU yields consistently higher correlations with hu-

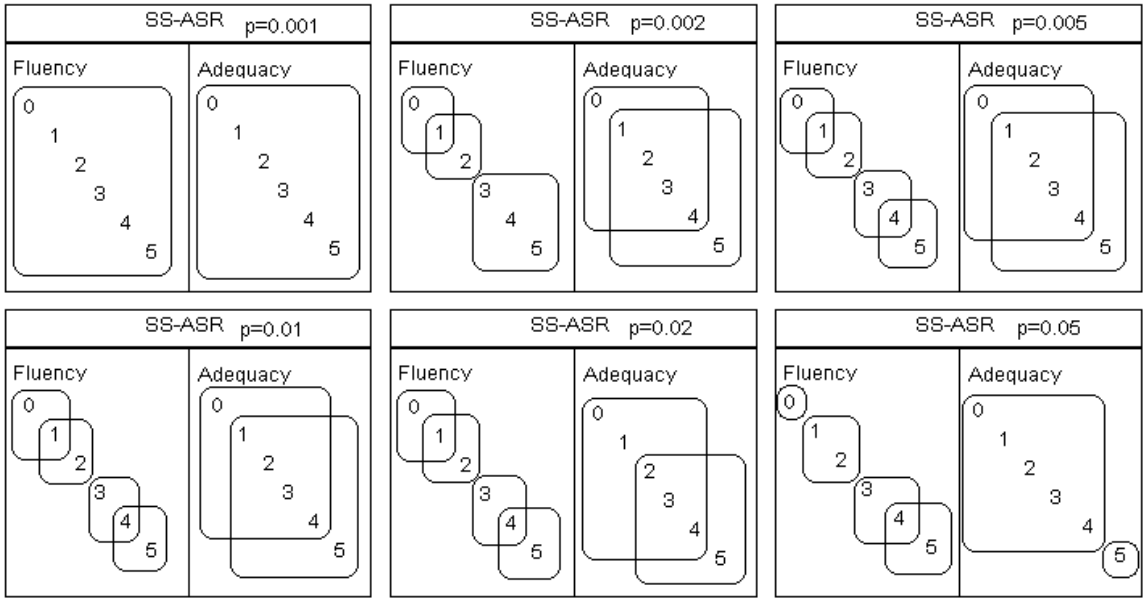


Figure 3: Clusterings obtained with diffrent p-values

man judgements of fluency, and GTM even obtains a negative score in the first input condition (spontaneous speech), showing a negative correlation with human ranking. In the case of adequacy, the picture is slightly less clear: BLEU seems to be more stable than the other metrics (it is better in two input conditions), even if METEOR has a higher correlation with adequacy in the Mixed Track. GTM-1 also achieves a high correlation for the Mixed Track. Let us also recall that this (indirect) approach based on the comparison of clusterings gives a view different from the computation of the direct correlation between segment-level or system-level hard rankings.[2]

We also compared how the clusterings obtained using the automatic evaluation metrics (BLEU, NIST, GTM-1, and METEOR) relate to each other. The results are displayed in Table 2.

Interestingly, the comparison scores between automatic evaluation metrics are higher than between the automatic evaluation metrics and the human judgement, which suggests that all these automatic metrics fall prey to some systematic error in evaluating translation quality.

[2]We do not claim that our method is better than direct correlation; instead it provides an alternative approach which is suited to the situation when an automatic metric is used to compare multiple systems.

	BLEU	NIST	METEOR
NIST	0.64	-	-
METEOR	0.77	0.79	-
GTM	0.70	0.79	0.86

Table 2: Comparing Automatic Metrics (Mixed Track)

4.4 Influence of the Significance Level

In Tables 1 and 2, the significance level is set to 0.05, since it is quite common to use such a value. However, this value affects the clusterings we obtain using our method (see e.g. Figure 3). In particular, a very small p-value (such as 0.001) yields inevitably a unique cluster containing all the systems, independently of the metric, which results in a correlation of 1 when comparing any two metrics. Obviously, there is a clear trade-off between the ability to produce a ranking and the level of confidence about this ranking.

In order to quantify the influence of this parameter, we compute the correlation between automatic and human evaluations, with various values of p. The results we obtain are displayed in Figure 4 for fluency and in Figure 5 for adequacy.

In terms of correlations with human judgements of fluency, the order of the automatic evaluation metrics does not seem to depend on signif-

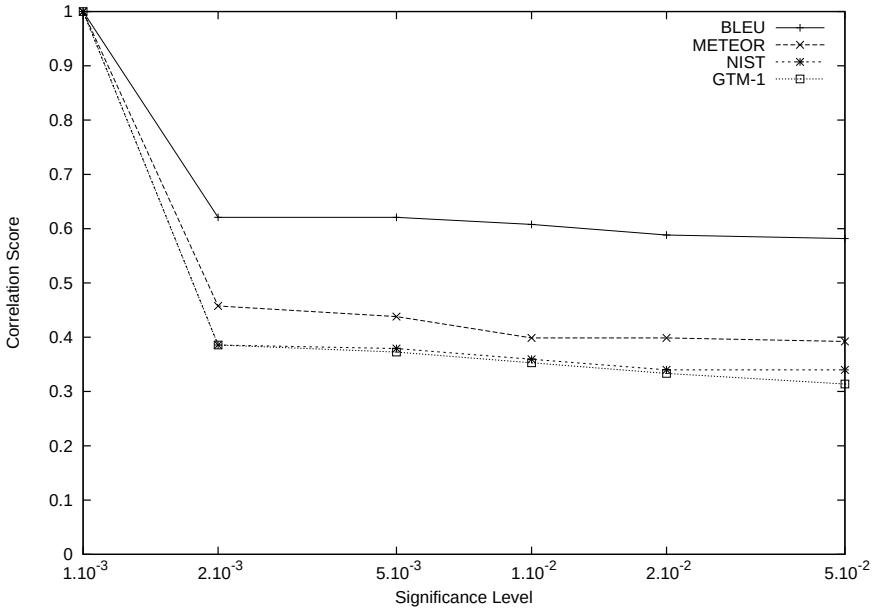


Figure 4: Influence of the p-value on the correlation with human judgements of fluency

icance level, and there is little variation between $p = 0.002$ and $p = 0.05$, although a very gentle decreasing trend can be noticed. Consequently, in this case, the choice of a p-value does not appear to be crucial. We can clearly observe that BLEU achieves the highest correlation with human judgements of fluency by a large margin.

Concerning adequacy, there is again little variation between $p = 0.002$ and $p = 0.05$, even if the relative order of the various metrics is not as stable. However, it seems that METEOR and GTM-1 are consistently better than the two other metrics, at least until $p = 0.05$.

5 Discussion and Conclusion

The variation in the number of clusters between tables in Figure 3 confirms the intuition that as the level of required confidence increases, it becomes more and more difficult to distinguish between different systems. The number of clusters ranges from one at $p = 0.001$, where all systems are seen as equal and the null hypothesis cannot be disproved, to four at $p = 0.05$ for fluency. Interestingly, clustering the systems with respect to their adequacy scores does not show the same level of refinement: at $p = 0.05$ there are only two (albeit non-overlapping) clusters. This tendency is not surprising, given that adequacy and fluency are two separate dimensions of a translation, each with its own set of conditions, so it is possible for systems to differ in the fluency of their output while being similar with respect to the semantic/lexical content. This duality of evaluation is often ignored in the creation of new automatic metrics for MT evaluation, where the guiding factor is usually the metric's correlation with the *average* human judgement.[3]

The comparison of clusters produced by BLEU, NIST, GTM, and METEOR on one hand, and human scores on the other, presented in Table 1, provides some surprising results. It turns out that BLEU, despite being widely criticised for low correlations with human judgements on segment level (Callison-Burch et al., 2006), consistently produces the most reliable clusters on the system level when it comes to judgements of fluency, and this trend is not influenced by the required significance level. Since BLEU was developed with system-level evaluation in mind, this is understandable; what is interesting, though, is that NIST, GTM, and METEOR, which are supposed to produce better segment-level evaluation than BLEU, are much worse than BLEU at

[3]Perhaps this is the reason why automatic metrics still seem so far away from successfully modeling human evaluation; it would be interesting to see whether we could devise a better metric by focusing on the two dimensions of fluency and adequacy separately.

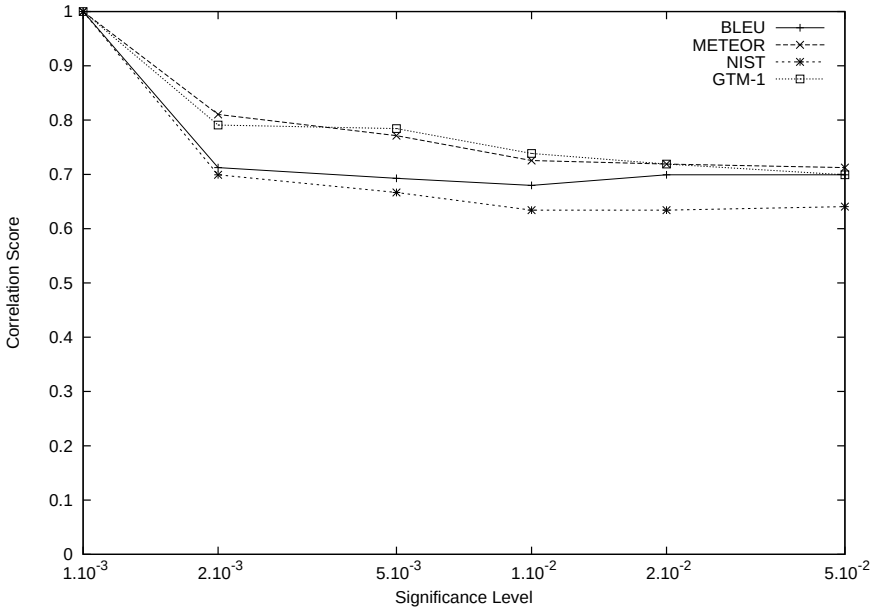


Figure 5: Influence of the p-value on the correlation with human judgements of adequacy

the system level - after all, we would expect the system-level evaluation to be directly dependent on the evaluation of its segments. This emphasizes the need to carefully choose one's metric depending on the type of task: it seems that for multiple system comparison BLEU does rather well, even though NIST, GTM, and METEOR might be more useful in the process of developing a single system (where the improvements often relate to specific types of sentences or structures and therefore a metric with a higher segment-level reliability would be better).

When it comes to correlations with human judgements of adequacy, these are on the whole higher for all the metrics; however, it must be remebered that the clusterings in the dimension of adequacy showed a much lower granularity than fluency, so it is easier to achieve high correlation. The difference between fluency and adequacy is smallest for BLEU, showing that a BLEU score reflects adequacy and fluency more equally than others. However, here BLEU is outperformed by METEOR and GTM-1, as the clusterings produced by these two metrics better reflect clusterings based on human judgement, at least for most values of p examined here. It seems then that here is where the advantage brought by better segment-level correlation with human judgement of METEOR and GTM is revealed.

Our future work includes conducting the clustering tests with a larger number of MT systems, to see whether the trends mentioned above hold in situations with a greater number of clusters. We also plan to add more metrics to our comparison, and vary the test with respect to the number of references available to the automatic metrics. Additionaly, we would like to compare the clusterings achieven in approximate randomization experiments with clusterings produced by a bootstrapping method for the same set of data.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the the ACL Workshop on Intrisic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 65–72, Ann Arbor, MI.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *Proceedings of EACL 2006*, pages 249–256, Trento, Italy.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL 2005*, pages 531–540, Ann Arbor, MI.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccur-

rence statistics. In *Proceedings of HLT 2002*, pages 128–132, San Diego, CA.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain.

I. Dan Melamed, Ryan Green, and Joseph P. Turian. 2004. Precision and recall of machine translation. In *Proceedings of HLT-NAACL 2003*, volume 2, pages 61–63, Edmonton, Canada.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience, New York, NY.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318, Philadelphia, PA.

Michael Paul. 2006. Overview of the IWSLT 2006 Evaluation Campaign. In *Proceedings of IWSLT 2006*, pages 1–15, Kyoto, Japan.

Stefan Riezler and John Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the the ACL Workshop on Intrisic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 57–64, Ann Arbor, MI.

T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of LREC 2002*, pages 147–152, Las Palmas, Spain.

Exploiting Source Similarity for SMT using Context-Informed Features

Nicolas Stroppa
Dublin City University,
Dublin,
Ireland
nstroppa@computing.dcu.ie

Antal van den Bosch
Tilburg University,
Tilburg,
The Netherlands
Antal.vdnBosch@uvt.nl

Andy Way
Dublin City University,
Dublin,
Ireland
away@computing.dcu.ie

Abstract

In this paper, we introduce context-informed features in a log-linear phrase-based SMT framework; these features enable us to exploit source similarity in addition to target similarity modeled by the language model. We present a memory-based classification framework that enables the estimation of these features while avoiding sparseness problems. We evaluate the performance of our approach on Italian-to-English and Chinese-to-English translation tasks using a state-of-the-art phrase-based SMT system, and report significant improvements for both BLEU and NIST scores when adding the context-informed features.

1 Introduction

In log-linear phrase-based SMT, the probability $\mathbb{P}(e_1^I|f_1^J)$ of target phrase e_1^I given a source phrase f_1^J is modeled as a (log-linear) combination of features that usually comprise some translational features, and a language model (Och and Ney, 2002). The usual translational features involved in those models express dependencies between source and target phrases, but not dependencies between source phrases themselves. In particular, the context in which those phrases occur is never taken into account during translation. While the language model can be seen as a way to exploit *target similarity* (between the translation and other target sentences), one could ask whether it is also possible to exploit *source similarity*, i.e. to take into account the context in which the source phrases to be translated actually occur.

In this paper, we introduce context-informed features in the original log-linear model, enabling us to take the context of source phrases into account during translation. In order to tackle the problems related to the estimation of these features, we propose a framework based on a memory-based classifier, which performs implicit smoothing. We also show that the addition of context-informed features, i.e. the source-similarity exploitation, results in an improvement in translation quality, for Italian-to-English and Chinese-to-English translations tasks.

2 Log-Linear Phrase-Based SMT

In statistical machine translation (SMT), translation is modeled as a decision process, in which the translation $e_1^I = e_1 \dots e_i \dots e_I$ of a source sentence $f_1^J = f_1^J = f_1 \dots f_j \dots f_J$ is chosen to maximize:

$$\underset{I,e_1^I}{\operatorname{argmax}}\, \mathbb{P}(e_1^I|f_1^J) = \underset{I,e_1^I}{\operatorname{argmax}}\, \mathbb{P}(f_1^J|e_1^I).\mathbb{P}(e_1^I), \quad (1)$$

where $\mathbb{P}(f_1^J|e_1^I)$ and $\mathbb{P}(e_1^I)$ denote respectively the translation model and the target language model (Brown et al., 1993). In log-linear phrase-based SMT, the posterior probability $\mathbb{P}(e_1^I|f_1^J)$ is directly modeled as a (log-linear) combination of features (Och and Ney, 2002), that usually comprise M translational features, and the language

model:

$$\log \mathbb{P}(e_1^I|f_1^J) = \sum_{m=1}^{m} \lambda_m h_m(f_1^J, e_1^I, s_1^K) + \lambda_{LM} \log \mathbb{P}(e_1^I), \quad (2)$$

where $s_1^K = s_1 \dots s_k$ denotes a segmentation of the source and target sentences respectively into the sequences of phrases $(\tilde{e_1}, \dots, \tilde{e_k})$ and $(\tilde{f_1}, \dots, \tilde{f_k})$ such that (we set $i_0 := 0$):

$$\forall 1 \leq k \leq K, \quad s_k := (i_k; b_k, j_k),$$
$$\tilde{e_k} := e_{i_{k-1}+1} \dots e_{i_k},$$
$$\tilde{f_k} := f_{b_k} \dots f_{j_k}.$$

A remarkable property of this approach is that the usual translational features involved in those models only depend on a pair of source/target phrases, i.e. they do not take into account the contexts of those phrases. This means that each feature h_m in equation (2) can be rewritten as:

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \tilde{h}_m(\tilde{f_k}, \tilde{e_k}, s_k), \quad (3)$$

where $\tilde{h}_m$ is a feature that applies to a single phrase-pair.[1] It thus follows:

$$\sum_{m=1}^{m} \lambda_m \sum_{k=1}^{K} \tilde{h}_m(\tilde{f_k}, \tilde{e_k}, s_k) = \sum_{k=1}^{K} \tilde{h}(\tilde{f_k}, \tilde{e_k}, s_k),$$
$$\text{with } \tilde{h} = \sum_{m=1}^{m} \lambda_m \tilde{h}_m. \quad (4)$$

In this context, the translation process amounts to: (i) choosing a segmentation of the source sentence, (ii) translating each source phrase, and possibly (iii) re-ordering the target segments obtained. The target language model is used to guide the decision process; in case no particular constraints are assumed, it is common to employ beam search techniques to reduce the number of hypotheses to be considered (Koehn, 2004). Equations (2) and (4) characterize what is referred to as the *standard phrase-based approach* in the following.

[1] Here, for notational purposes, we exclude re-ordering features that might not be expressed using equation (3). This does not affect our general line of reasoning.

C'è una partita di baseball oggi ?
(⇔ *Is there a baseball game today?*)

– Possible translations for *partita*:

game	*partita di calcio* ⇔ *a soccer game*
gone	*è partita* ⇔ *she has gone*
partita	*una partita di Bach* ⇔ *a partita of Bach*

– Possible translations for *di*:

of	*una tazza di caffè* ⇔ *a cup of coffee*
	prima di partire ⇔ *before coming*

Figure 1: Examples of ambiguity for the (Italian) word *partita*, easily solved when considering its context

3 Context-Informed Features

3.1 Context-Based Disambiguation

The optimization of the feature weights λ_m can be performed in a *discriminative* learning setting (Och and Ney, 2002). However, it is important to note that these weights are *meta-parameters*. Indeed, the dependencies between the parameters of the standard phrase-based approach consist of: (i) relationships between single phrases (modeled by $\tilde{h}$), (ii) relationships between consecutive target words (modeled by the language model), which is generally characteristic of *generative* models (Collins, 2002; Dietterich, 2002). Notably, dependencies between consecutive *source* phrases are not directly expressed.

Discriminative frameworks usually allow for the introduction of (relatively) unrestricted dependencies that are relevant to the decision process. In particular, disambiguation problems can be solved by taking the direct context of the entity to disambiguate into account (e.g. Dietterich (2002)). In the translation example displayed in Figure 1, the source right context is sufficient to solve the ambiguity: when followed by *di baseball*, the (Italian) word *partita* is very likely to correspond to the (English) word *game*.

However, in the standard phrase-based approach, the disambiguation strongly relies on the *target* language model. Indeed, even though the various translation features associated with *partita* and *game*, *partita* and *gone*, etc., may depend on the type of data on which the model is trained, it is likely that most language models will select the correct translation *baseball game* as the most

probable among all the possible combinations of target words: *gone of baseball*, *game of baseball*, *baseball partita*, *baseball game*, etc., but this solution appears to be more expensive than simply looking at the context. In particular, the context can be used to early prune weak candidates, which allows spending more time on promising candidates.

Several discriminative frameworks have been proposed recently in the context of MT to fully exploit the flexibility of discriminative approaches (Cowan et al., 2006; Liang et al., 2006; Tillmann and Zhang, 2006; Wellington et al., 2006). Unfortunately, this flexibility usually comes at the price of training complexity. An alternative in-between approach, pursued in this paper, consists of introducing context-informed features in the original log-linear framework. This enables us to take the context of source phrases into accounts, while benefiting from the existing training and optimization procedures of the standard phrase-based approach.

3.2 Context-Informed Features

In this Section, we introduce several features that take the context of source phrases into account.

Word-based features A feature that includes the direct left and right context words (resp. f_{b_k-1} and f_{j_k+1}) of a given phrase $\tilde{f}_k = f_{b_k} \dots f_{j_k}$ takes the following form:

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \tilde{h}_m(\tilde{f}_k, f_{b_k-1}, f_{j_k+1}, \tilde{e_k}, s_k).$$

In this case, the contextual information can be seen as a window of size 3 (focus phrase + left context word + right context word), centered on the source phrase $\tilde{f}_k$. Larger contexts may also be considered. More generally, we have:

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \tilde{h}_m(\tilde{f}_k, CI(\tilde{f}_k), \tilde{e_k}, s_k),$$

where $CI(\tilde{f}_k)$ denotes some contextual information about $\tilde{f}_k$.[2]

[2]The definition of the context may be language dependent. For example, one could consider only the right context if it makes sense to do so for a particular language; the same remark holds for the size of the context.

Class-based features In addition to the context words themselves, it is possible to exploit several knowledge sources characterizing the context. For example, we can consider the Part-Of-Speech of the focus phrase and of the context words.[3] In this case, the contextual information takes the following form for a window of size 3:

$$CI(\tilde{f}_k) = \langle POS(\tilde{f}_k), POS(f_{b_k-1}), POS(f_{j_k+1}) \rangle.$$

We can also combine the class-based and the word-based information.

Feature definition One natural definition to express a context-informed feature consists of viewing it as the conditional probability of the target phrase given the source phrase and its context information:

$$\tilde{h}_m(\tilde{f}_k, CI(\tilde{f}_k), \tilde{e_k}, s_k) = \log \mathbb{P}(\tilde{e_k} | \tilde{f}_k, CI(\tilde{f}_k)).$$

The problems related to the estimation of these probabilities are addressed in the next section.

4 Memory-Based Disambiguation

4.1 A Classification Approach

The direct estimation of $\mathbb{P}(\tilde{e_k} | \tilde{f}_k, CI(\tilde{f}_k))$, for example using relative frequencies, is problematic. Indeed, it is well known that the estimation of $\mathbb{P}(\tilde{e_k} | \tilde{f}_k)$ using relative frequencies results in the overestimation of the probabilities of long phrases (Zens and Ney, 2004; Foster et al., 2006); a frequent remedy consists of introducing a smoothing factor, which takes the form of lexical-based features (Zens and Ney, 2004). Similar issues and a variety of smoothing techniques are discussed in (Foster et al., 2006). In the case of context-informed features, since the context is also taken into account, this estimation problem can only worsen, which forbids us to use relative frequencies.

To avoid these issues, we use a memory-based classifier, which enables *implicit smoothing*. More precisely, in order to estimate the probability $\mathbb{P}(\tilde{e_k} | \tilde{f}_k, CI(\tilde{f}_k))$, we ask a memory-based classifier to classify the input $\langle \tilde{f}_k, CI(\tilde{f}_k) \rangle$ (seen

[3]The POS of a multi-word focus phrase is the concatenation the POS of the words composing the phrase.

as a fixed-length vector). The result of this classification is a set of weighted class labels, representing the possible target phrases $\tilde{e_k}$. Once normalized, these weights can be seen as the posterior probabilities of the target phrases $\tilde{e_k}$, which thus gives access to $\mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$.

In order to build the set of examples required to train the classifier, we slightly modify the standard phrase extraction procedure described in (Koehn et al., 2003) so that it also extracts the context information of the source phrases; since these aligned phrases are needed in the standard phrase-based approach, the context extraction comes at no additional cost.

Note that there are several reasons for using a memory-based classifier: (i) training can be performed efficiently, even with millions of examples, (ii) it is insensitive to the number of output classes, (iii) its output can be seen as a posterior distribution.

4.2 IGTree Classification

In the following, we describe IGTREE,[4] an algorithm for the top-down induction of decision trees that can be seen as an approximation of 1-nearest neighbor that stores and classifies examples efficiently (Daelemans et al., 1997). IGTREE compresses a database of labeled examples into a lossless-compression decision-tree structure that preserves the labeling information of all examples (and technically should be named a *trie* according to Knuth (1973)). In our case, a labeled example is a fixed-length feature-value vector representing the source phrase and its contextual information, associated with a symbolic class label representing the associated target phrase. The trie that is constructed can then be used to predict a target phrase given a source phrase and its context. A typical trie is composed of nodes that each represent a partition of the original example database, together with the most frequent class of that partition. The root node of the trie thus represents the entire example database and carries the most frequent value as class label, while end nodes (leaves) represent a homogeneous partition of the database in which all examples have the same class label. A node is either a leaf or a non-ending node that branches out to nodes at a deeper level of the trie. Each branch represents a test on a feature value; branches fanning out of one node test on values of the same feature.

Prediction in IGTREE is a straightforward traversal of the trie from the root node down, where a step is triggered by an exact match between a feature of the new example and an arc fanning out of the current node. When the next step ends in a leaf node, the homogeneous class at that node is returned; when no match is found with an arc fanning out of the current node, the most likely class stored at that node is returned.

To attain high compression levels, IGTREE adopts the same heuristic that most other decision-tree induction algorithms adopt, such as C4.5 (Quinlan, 1983), which is to create trees from a starting root node and branch out to test on the most informative, or most class-discriminative features first. Like C4.5, IGTREE uses information gain (IG) to estimate the discriminative power of features. The key difference between IGTREE and C4.5 is that IGTREE computes the IG of all features once on the full database of training examples, makes a feature ordering once on these computed IG values, and uses this ordering throughout the whole trie. Moreover, IGTREE does not prune its produced trie, so that it performs a lossless compression of the labeling information of the original example database. In case of exact matches, the exact same output will be retrieved.

IGTREE bases its classification on the example that matches on most features, ordered by their IG, and guesses a majority class of the set of examples represented at the level of mismatching. In our case, we do not keep just the majority class since we want to be able to estimate $\mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$ for all possible $\tilde{e_k}$; we are thus interested in the entire set of labels represented at the level of mismatching. Each possible target phrase can be supported by multiple votes, which leads to a weighted set of target phrases. By normalizing these weights, we obtain the posterior probability distributions we are interested in.[5]

[4] An implementation of IGTree is freely available as part of the TiMBL software package, which can be downloaded from `http://ilk.uvt.nl/timbl`.

[5] It is also interesting to note that if we do not include any context information, the (normalized) output provided by IGTREE exactly corresponds to the conditional probabil-

4.3 Memory-Based Features

The weighted set of possible target phrases given a source phrase and its context is an intermediary result of the estimation of $\mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$. In addition to the feature $\tilde{h}_m(\tilde{f_k}, CI(\tilde{f_k}), \tilde{e_k}, s_k) = \log \mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$, we consider a simple binary feature based on this intermediary result:

$$\tilde{h}_{best} = \begin{cases} 1 & \text{if } \tilde{e_k} \text{ is (one of) the target phrases} \\ & \text{with the most support,} \\ 0 & \text{otherwise,} \end{cases}$$

where "most support" means the highest probability according to $\mathbb{P}(\tilde{e_k}|\tilde{f_k}, CI(\tilde{f_k}))$. The two features $\tilde{h}_m$ and $\tilde{h}_{best}$ are integrated in the log-linear model. As for the standard phrase-based approach, their weights are optimized using minimum-error-rate training (Och, 2003).

4.4 Implementation Issues

When predicting a target phrase given a source phrase and its context, the source phrase is intuitively the feature with the highest prediction power; in all our experiments, it is the feature with the highest IG. In the trie constructed by IGTREE, this is thus the feature on which the first branching decision is taken. Consequently, when classifying a source phrase $\tilde{f_k}$ with its context, there are two possible situations, depending on $\tilde{f_k}$ being in the training material or not. In the first case, $\tilde{f_k}$ is matched, and we proceed further down the trie. At this stage, it follows that the target phrases that can be retrieved are only those that have been aligned to $\tilde{f_k}$. In the second case, $\tilde{f_k}$ cannot be matched, so the full set of labeled leaves of the entire trie is retrieved. Since the second case does not present any interest, we limit the classification to the source phrases contained in the training material. By limiting ourselves to the first situation, we ensure that only target phrases $\tilde{e_k}$ that have been aligned with $\tilde{f_k}$ will be retrieved. This is a desirable property that may be not be necessarily verified if we were using a different type of classifier, more prone to over-generalisation issues.[6]

Phrase-based SMT decoders such as (Koehn, 2004) rely on a phrase-table represented as a list of aligned phrases accompanied with several features. Since these features do not express the context in which those phrases occur, no context information is kept in the phrase-table, and there is no way to recover this information from the phrase-table. In order to take into account the context-informed features with this kind of decoders, we use the workaround described in what follows. Each word to be translated (i.e. appearing in the test set) is assigned a unique id, and each phrase to be translated which is also present in the phrase-table is given to IGTREE for classification. We merge the initial information of the phrase-table concerning this source phrase with the output for IGTREE, to obtain a new phrase-table containing the standard and the context-informed features. In this new phrase-table, each source phrase is represented as a sequence of ids (of the words composing the phrase). By replacing all the words by their ids in the test set, we can translate it using this new phrase-table.

4.5 Source vs. Target Similarity

SMT and target-based similarity The probability of a (target) sentence with respect to a n-gram-based language model can be seen as a measure of similarity between this sentence and the sentences found in the corpus C on which the language model is trained. Indeed, the language model will assign high probabilities to those sentences which share lots of n-grams with the sentences of C, while sentences with few n-grams matches will be assigned low probabilities. In other words, the language model is used to make the resulting translation similar to previously seen (target) sentences: SMT is *target-similarity* based.

EBMT and source-based similarity In order to perform the translation of a given sentence f, Example-Based Machine Translation (EBMT) systems (i) look for source sentences similar to f in the bilingual corpus (retrieval), (ii) find use-

ities $\mathbb{P}(\tilde{e_k}|\tilde{f_k})$ estimated with relative frequencies on the set of aligned phrases.

[6] From the point of view of the classification task, the set of class labels is the set of *all* the target phrases encountered in the training data. Consequently, given a source phrase $\tilde{f_k}$ there is in the general case nothing preventing a classifier to output a target phrase $\tilde{e_k}$ that was never aligned to $\tilde{f_k}$. If we use IGTree and if the source phrase is the feature with the highest information gain, then we have the mentioned desirable property.

ful fragments in these sentences (matching), (iii) adapts and recombine the translation of these fragments (transfer) (Nagao, 1984; Somers, 1999; Carl and Way, 2003). A number of matching techniques and notions of similarity have been proposed. Consequently, EBMT crucially relies on the retrieval of *source* sentences *similar* to f in the bilingual training corpus; in other words, EBMT is *source-similarity* based. Let us also mention (Somers et al., 1994), which marks the fragments to translate with their (left and right) contexts.

Source and Target Similarity While the use of target-similarity may avoid problems such as boundary-friction usually encountered in EBMT (Brown et al., 2003), the use of source-similarity may limit ambiguity problems (cf. Section 3). By exploiting the two types of similarity, we hope to benefit from the strength of both aspects.

5 Experimental Results

5.1 Data, Tasks, and Baseline

The experiments were carried out using the Chinese–English and Italian–English datasets provided within the IWSLT 2006 evaluation campaign (Paul, 2006), extracted from the Basic Travel Expression Corpus (BTEC) (Takezawa et al., 2002). This multilingual speech corpus contains sentences similar to those that are usually found in phrase-books for tourists going abroad. Training was performed using the default training set, to which we added the sets devset1, devset2, and devset3. The development set (devset 4) was used for tuning purposes (in particular for the optimisation of the weights of the log-linear model), and the final evaluation is conducted using the test set (using the CRR=Correct Recognition Result input condition). For both Chinese and Italian, POS-tagging is performed using the MXPOST tagger (Ratnaparkhi, 1996). Table 1 summarizes the various corpus statistics. The number of training/test examples refers to the examples involved in the classification task.

For all experiments, the quality of the translation output is evaluated using the accuracy measures BLEU (Papineni et al., 2002), NIST (Doddington, 2002), and METEOR (Banerjee and Lavie, 2005), using 7 references and ignoring case information. For BLEU and NIST, we also report statistical significance p-values, estimated using approximate randomization (Noreen, 1989; Riezler and Maxwell, 2005).[7]

	Chinese–English		Italian–English	
Train.				
Sentences	44,501		21,484	
Running words	323,958	351,303	156,237	169,476
Vocabulary size	11,421	10,363	10,418	7,359
Train. examples	434,442		391,626	
Dev.				
Sentences	489 (7 refs.)		489 (7 refs.)	
Running words	5,214	39,183	4,976	39,368
Vocabulary size	1,137	1,821	1,234	1,776
Test examples	8,004		7,993	
Eval.				
Sentences	500 (7 refs.)		500 (7 refs.)	
Running words	5,550	44,089	5,787	44,271
Vocabulary size	1,328	2,038	1,467	1,976
Test examples	8,301		9,103	

Table 1: Chinese–English and Italian–English corpus statistics

To assess the validity of our approach, we use the state-of-the-art phrase-based SMT system MOSES (Koehn et al., 2007).[8] The baseline system is composed of the usual features: phrase-based probabilities and lexical weighting in both directions, phrase and word penalties, and re-ordering. Our system additionally includes the memory-based features described in Sections 3 and 4.

5.2 Translation Results

The results obtained for the Italian–English and Chinese–English translation tasks using the IWSLT data are summarized in Table 2. The contextual information may include the (context) words, their Part-Of-Speech, or both, respectively denoted by Words-only, POS-only, and Words+POS in the following. In all cases, the size of the left context is 2 and so is the size of the right context.[9]

In the case of Italian–English, a consistent improvement is observed for all metrics, for the three types of contextual information (Words-only, POS-only, Words+POS). Relatively to the baseline results, this improvement is significant

[7] The code for statistical significance testing can be freely downloaded from `http://www.computing.dcu.ie/~nstroppa/softs/fast_mt_eval.tgz`.

[8] `http://www.statmt.org/moses/`

[9] These are the values which led to the best results on the development set during the exploratory phase.

	BLEU[%] (p-value)	NIST (p-value)	METEOR[%]
Italian–English			
Baseline	37.84	8.33	65.63
POS-only	**38.56** (< 0.1)	8.45 (< 0.02)	66.03
Words-only	37.93 ($\times$)	8.43 (< 0.02)	66.11
Words+POS	38.12 ($\times$)	**8.46** (< 0.01)	**66.14**
Chinese–English			
Baseline	18.81	5.95	47.17
POS-only	19.64 (< 0.005)	6.10 (< 0.005)	47.82
Words-only	**19.86** (< 0.02)	**6.23** (< 0.002)	**48.34**
Words+POS	19.19 ($\times$)	6.09 (< 0.005)	47.97

Table 2: Italian–English and Chinese–English Translation Results

for NIST, and marginally significant for BLEU (p-value < 0.1) for POS-only. The combination of the words and POS information leads to a slight improvement for NIST and METEOR relatively to Words-only and POS-only. As for the BLEU score, the best results are obtained with POS-only. The difference between POS-only, Word-only, and Words+POS is never statistically significant. The difference of significance between the BLEU and NIST scores is investigated in more depth in Section 5.3.

In the case of Chinese–English, the improvement is also consistent for all metrics, and significant for both BLEU and NIST for Words-only, POS-only, and Words+POS. Interestingly, the addition of Part-of-Speech information does not seem to be benefitial in the case of Chinese. Indeed, the results of Words-only are higher than those obtained with both POS-only and Words+POS. In order to understand better why this is the case, we manually inspected the tagger's output for the Chinese data. The most obvious explanation is simply the (poor) quality of tagging. Indeed, we found lots of tagging mistakes, which contributes to the introduction of noise in the data. We also manually checked that in the case of Italian, the tagging accuracy is qualitatively higher. Consequently, even if there is something to be gained from the addition of POS information, it seems important to ensure that the accuracy of tagging is high enough. Also, with larger training data, it may be sufficient to rely on the words only, since the need for generalization is less important in this case.

In order to know the contribution of the various contextual elements, we rank the contextual features of the Words+POS model based on their Information Gain (cf. Table 3). W(0) and P(0) denotes the focus phrase and its POS, while W(i) and P(i) denotes the word and the POS of the words at position i relative to the focus phrase. The rankings for Italian and Chinese are globally

	Italian–English		Chinese–English	
Rank	Feature	IG	Feature	IG
1	W(0)	7.82	W(0)	6.74
2	P(0)	4.59	W(+1)	3.73
3	W(+1)	4.24	P(0)	3.23
4	W(-1)	4.09	W(-1)	3.21
5	W(+2)	3.19	W(+2)	2.90
6	W(-2)	2.84	W(-2)	2.25
7	P(+1)	1.75	P(-1)	1.18
8	P(-1)	1.61	P(+1)	1.03
9	P(-2)	0.94	P(-2)	0.77
10	P(+2)	0.90	P(+2)	0.75

Table 3: Feature Information Gain

similar, and we can observe the following tendencies:

Word information $>$ POS information,
Focus $>$ Right context $>$ Left context.

5.3 Statistical Significance for n-gram Based Metrics

Since the BLEU and NIST metrics are both precision- and n-gram-based (Doddington, 2002), it is somehow strange that an improvement may be statistically significant for NIST and insignificant for BLEU (as it is the case 3 times in Table 2). The differences between the two metrics are: (i) the maximum length of the n-gram considered (4 for BLEU, 5 for NIST), (ii) the weighting of the matched n-grams

(no weighting for BLEU, information-based weighting for NIST), (iii) the type of mean used to aggregate the number of matched n-grams for different n (geometric for BLEU, arithmetic for NIST), (iv) the length penalty.

To test which of these options were responsible for the difference in significance, we created the 2^4 metrics corresponding to all the possible combinations of options, and we ran the significance tests for the three cases for which there was a disagreement between BLEU and NIST with respect to significance. We found out that the most important factors are the information-based weighting, and the type of mean used. This is actually consistent with our expectation for our system regarding lexical selection. Indeed, BLEU's geometric mean tends to ignore good lexical changes, which may be shadowed by low n-grams results for high values of n; similarly, the information-based weighting favors the most difficult lexical choices. Note that these remarks are also consistent with the findings of (Riezler and Maxwell, 2005).

6 Related Work

Several proposals have been recently made to fully exploit the accuracy and the flexibility of discriminative learning (Cowan et al., 2006; Liang et al., 2006; Tillmann and Zhang, 2006; Wellington et al., 2006). These papers generally require one to redefine one's training procedures; on the contrary our approach introduces new features while keeping the strength of existing state-of-the-art systems. The exploitation of source-similarity is one of the key components of EBMT (Nagao, 1984; Somers, 1999; Carl and Way, 2003); one could say that our approach is a combination of EBMT and SMT since we exploit both source similarity and target similarity. (Carpuat and Wu, 2005) present an attempt to use word-sense disambiguation techniques to MT in order to enhance lexical selection; in a sense, we are also performing some sort of word-sense disambiguation, even if the handling of lexical selection is performed totally implicitly in our case.

7 Conclusion

In this paper, we have introduced new features for log-linear phrase-based SMT, that take into account contextual information about the source phrases to translate. This contextual information can take the form of left and right context words, as well as other source of knowledge such as Part-Of-Speech information. We presented a memory-based classification framework that enables the estimation of these features while avoiding sparseness problems.

We have evaluated the performance of our approach by measuring the influence of the addition of these context-informed features on Italian-to-English and Chinese-to-English translation tasks, using a state-of-the-art phrase-based SMT system. We report significant improvements for both BLEU and NIST scores.

As for future work, we plan to investigate the addition of features including syntactic information. For example, one could consider dependency relationships between the words within the focus (source) phrase or with its close context. We could also introduce context-informed lexical smoothing features, similarly to the standard phrase-based approach. Finally, we plan to modify the decoder to directly integrate context-informed features.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the the ACL Workshop on Intrisic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 65–72, Ann Arbor, MI.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Ralf D. Brown, Rebecca Hutchinson, Paul N. Bennett, Jaime G. Carbonell, and Peter Jansen. 2003. Reducing boundary friction using translation-fragment overlap. In *Proceedings of the 9th Machine Translation Summit*, pages 24–31, New Orleans, LA.

Michael Carl and Andy Way, editors. 2003. *Recent Advances in Example-Based Machine Translation*, volume 21 of *Text, Speech and Language Technology*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Marine Carpuat and Dekai Wu. 2005. Word sense disambiguation vs. statistical machine translation. In *Proceedings of ACL 2005*, pages 387–394, Ann Arbor, MI.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of EMNLP 2002*, pages 1–8, Philadelphia, PA.

Brooke Cowan, Ivona Kucerovà, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proceedings of EMNLP 2006*, pages 232–241, Sydney, Australia.

Walter Daelemans, Antal Van den Bosch, and A. Weijters. 1997. IGTree: using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423.

Thomas G. Dietterich. 2002. Machine learning for sequential data: A review. In Terry Caelli, Adnan Amin, Robert P. W. Duin, Mohamed S. Kamel, and Dick de Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*, pages 15–30. Springer-Verlag.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of HLT 2002*, pages 128–132, San Diego, CA.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of EMNLP 2006*, pages 53–61, Sydney, Australia.

Donald E. Knuth. 1973. *The art of computer programming*, volume 3: Sorting and searching. Addison-Wesley, Reading, MA.

Philip Koehn, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 48–54, Edmonton, Canada.

P. Koehn, M. Federico, W. Shen, N. Bartoldi, O. Bojar, C. Callison-Burch, B. Cowan, C. Dyer, H. Hoang, R. Zens, A. Constantin, C. Moran, and E. Herbst. 2007. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding. Technical report, Final Report of the Johns Hopkins 2006 Summer Workshop.

Philip Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of AMTA 2004*, pages 115–124, Washington, DC.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of COLING-ACL 2006*, pages 761–768, Sydney, Australia.

Makoto Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In Alick Elithorn and Ranan Banerji, editors, *Artificial and Human Intelligence*, pages 173–180, Amsterdam, The Netherlands. North-Holland.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience, New York, NY.

Franz Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL 2002*, pages 295–302, Philadelphia, PA.

Franz Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*, pages 160–167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318, Philadelphia, PA.

Michael Paul. 2006. Overview of the IWSLT 2006 Evaluation Campaign. In *Proceedings of IWSLT 2006*, pages 1–15, Kyoto, Japan.

Ross Quinlan. 1983. Learning efficient classification procedures and their application to chess endgames. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 463–482. Morgan Kaufmann Publishers, Los Altos, CA.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP 1996*, pages 133–142, Philadelphia, PA.

Stefan Riezler and John Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the the ACL Workshop on Intrisic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 57–64, Ann Arbor, MI.

Harold Somers, Ian McLean, and Danny Jones. 1994. Experiments in multilingual example-based generation. In *CSNLP 1994: 3rd Conference on the Cognitive Science of Natural Language Processing*, Dublin, Ireland.

Harold Somers. 1999. Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157.

T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of LREC 2002*, pages 147–152, Las Palmas, Spain.

Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical mt. In *Proceedings of COLING-ACL 2006*, pages 721–728, Sydney, Australia.

Benjamin Wellington, Joseph Turian, Chris Pike, and I. Dan Melamed. 2006. Scalable purely-discriminative training for word and tree transducers. In *Proceedings of AMTA 2006*, pages 251–260, Cambridge, MA.

Richard Zens and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. In *Proceedings of HLT-NAACL 2004*, pages 257–264, Boston, MA.

Phrase Alignment Based on Bilingual Parsing

Akira Ushioda

Software and Solution Laboratories
Fujitsu Laboratories Ltd.
4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki 211-8588
Japan
ushioda@jp.fujitsu.com

Abstract

A novel approach is presented for extracting syntactically motivated phrase alignments. In this method we can incorporate conventional resources such as dictionaries and grammar rules into a statistical optimization framework for phrase alignment. The method extracts bilingual phrases by incrementally merging adjacent words or phrases on both source and target language side in accordance with a global statistical metric. The extracted phrases achieve a maximum F-measure of over 80 with respect to human judged phrase alignments. The extracted phrases used as training corpus for a phrase-based SMT shows better cross-domain portability over conventional SMT framework.

1 Introduction

In the phrase-based SMT framework (Marcu & Wong, 2002; Och & Ney, 2004; Chiang, 2005), extraction of phrase pairs is a key issue. Currently the standard method of extracting bilingual phrases is to use a heuristics such as diag-and (Koehn et. al., 2003). In this method starting with the intersection of word alignments of both translation directions additional alignment points are added according to a number of heuristics and all the phrase pairs which are consistent with the word alignments are collected.

Although this method is effective by itself it is very difficult to incorporate syntactic information in a straight manner because phrases extracted by this method have basically little syntactic significance. Especially if we intend to combine strength of conventional rule-based approach with that of SMT, it is essential that phrases, or translation units, carry syntactic significance such as being a constituent (Yamada & Knight, 2001).

Another drawback of the conventional method is that the phrase extraction process is deterministic and no quantitative evaluation is applied. Furthermore if the initial word alignments have errors, these errors propagate to the phrase alignment process. In doing so the burden of statistical optimization is imposed on the final decoding process. We propose in this paper a novel phrase alignment method in which we can incorporate conventional resources such as dictionaries and grammar rules into a statistical optimization framework for phrase alignment.

The outline of the proposed method, applied to Japanese-English bilingual corpus, is as follows.

1) The training bilingual corpus is first word-aligned by GIZA++ (Och & Ney, 2000).
2) A word translation model is learnt by relative frequency from the word-alignment and smoothed by a bilingual dictionary.
3) Chunking is performed on both sides.
4) The probability that an English word belongs to a Japanese chunk is evaluated from which an entropy score is computed.
5) The entropy score is used to guide the process of merging adjacent phrases of both languages.
6) The merging process terminates when the score takes a minimum value.

Although the above steps are purely guided by a statistical metric, some syntactic preferences or constraints can guide the search.

The objective of this work is to extract alignments of phrases which are linguistically motivated. However, there is no guarantee that even manually extracting, out of aligned sentences, bilingual phrases which correspond to each other in meaning results in a collection of pairs of source and target phrases which are both constituents. There might be cases in which a phrase in one language constitutes a constituent while the corresponding phrase in the other language does not. Therefore the basic strategy we adopt here is to try to extract bilingual phrases whose source language side at least constitutes a constituent. As for the target language side, a preference is given to constituent constructs.

2 Phrase Alignment Method

The phrase alignment method we propose here extracts bilingual phrases by incrementally merging adjacent words or phrases on both source and target languages in accordance with a global statistical metric along with syntactic constraints and preferences.

The merging process is guided by an entropy score which is calculated from the *alignment matrix*. Figure 1 shows an example of the alignment matrix for the following sentence pair:

(1a) 演算回路の記憶値の乗算と新しいデータの加算のループを繰り返すことにより，簡単な演算回路で現在のデータに重みを置いた平均値を算出可能とする。

(1b) *To calculate an average value weighed in the present data with a simple arithmetic circuit by repeating the loop of multiplication of the stored value in the arithmetic circuit and the addition of a new data.*

In the alignment matrix, English words are arranged in each row and Japanese chunks are arranged in each column. The value of the (i, j) element divided by the margin of the i-th row represents the probability that the translation of the i-th English word (w_i) appears in the j-th Japanese chunk (j_j). For example, the translation of w_1 (calculate) can be "**演算**", which appears in j_0 ("**演算**回路の記憶値の") and j_8 ("簡単な**演算**回路で"), or "**算出**", which appears in j_{13} ("**算出**可能とする"), or "**算**" , which appears in j_1 and j_3 in addition to j_0, j_8 and j_{13}. Since "calculate" is more likely to be translated as "**算出**" than others, the (1, 13) element has larger value than other elements in the same row. Determiners, prepositions, conjunctions, and other function words are treated as stopwords and their elements are all assigned a value of zero. When there is more than one element with a positive value in the same row, these elements are shown in Figure 1 with a shaded square, and this means that the corresponding English word is ambiguous on the identity of the corresponding Japanese chunk. On the other hand, if there is only one element, say (p,q), with positive value in the same row, it is certain that the English word w_p belongs to the Japanese chunk j_q. If there is one and only one nonzero element in each row and in each column, then we have a complete one-to-one matching between Japanese elements (phrases) and English elements (words or phrases). The intuition behind the proposed method is that by merging adjacent elements which constitute a phrase and tend to stay together in both languages, the alignment matrix approaches a one-to-one matching. Therefore if there is a global measure that shows how close the current alignment matrix is to a one-to-one matching, we can use it to guide the merging process. We use the entropy score which is described in the next section.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 To
9	0.11	0	0.11	0	0	0	0	9	0	0	0	0	64	1 calculate
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2 an
0	0	0	0	0	0	0	0	0	0	0	0	85	0	3 average
96	0	0	0	0	0	0	0	0	0	0	0	96	0	4 value
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5 weighed
0	0	0	0	0	0	0	0	0	0	0	0	0	0	6 in
0	0	0	0	0	0	0	0	0	0	0	0	0	0	7 the
0	0	0	0	0	0	0	0	0	34	0	0	0	0.04	8 present
0	0	98	0	0	0	0	0	0	98	0	0	0	0	9 data
0	0	0	0	0	0	0	0	0	0	0	0	0	0	10 with
0	0	0	0	0	0	0	0	0	0	0	0	0	0	11 a
0	0	0	0	0	0	0	0	56	0	0	0	0	0	12 simple
97	0	0	0	0	0	0	0	97	0	0	0	0	0	13 arithmetic
99	0	0	0	0	0	0	0	99	0	0	0	0	0	14 circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	15 by
0	0	0	0	0	53	0	0	0	0	0	0	0	0	16 repeating
0	0	0	0	0	0	0	0	0	0	0	0	0	0	17 the
0	0	0	0	97	0	0	0	0	0	0	0	0	0	18 loop
0	0	0	0	0	0	0	0	0	0	0	0	0	0	19 of
0	72	0	0	0	0	0	0	0	0	0	0	0	0	20 multiplication
0	0	0	0	0	0	0	0	0	0	0	0	0	0	21 of
0	0	0	0	0	0	0	0	0	0	0	0	0	0	22 the
27	0	0	0	0	0	0	0	0	0	0	0	0	0	23 stored
96	0	0	0	0	0	0	0	0	0	0	0	96	0	24 value
0	0	0	0	0	0	0	0	0	0	0	0	0	0	25 in
0	0	0	0	0	0	0	0	0	0	0	0	0	0	26 the
97	0	0	0	0	0	0	0	97	0	0	0	0	0	27 arithmetic
99	0	0	0	0	0	0	0	99	0	0	0	0	0	28 circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	29 and
0	0	0	0	0	0	0	0	0	0	0	0	0	0	30 the
0	0	0	12	0	0	0	0	0	0	0	0	0	0	31 addition
0	0	0	0	0	0	0	0	0	0	0	0	0	0	32 of
0	0	0	0	0	0	0	0	0	0	0	0	0	0	33 a
0	0	20	0	0	0	0	0	0	0	0	0	0	0	34 new
0	0	98	0	0	0	0	0	0	98	0	0	0	0	35 data

[0]:演算回路の記憶値の
[1]:乗算と
[2]:新しいデータの
[3]:加算の
[4]:ループを
[5]:繰り返す
[6]:ことにより
[7]:,
[8]:簡単な演算回路で
[9]:現在のデータに
[10]:重みを
[11]:置いた
[12]:平均値を
[13]:算出可能とする

Figure 1: An example of the alignment matrix

2.1 Without Syntactic Information

We begin by describing the proposed phrase alignment method in the case of incorporating no syntactic information. Figure 2 shows the framework of the phrase aligner. In the case of incorporating no syntactic information, *Syntactic Component* in the figure plays no role. We take here an example of translating from Japanese to English, but the framework presented here basically works for any language pair as long as a conventional rule-based approach is applicable.

As a preparation step, word alignments are obtained from a bilingual corpus by GIZA++ for both directions (source to target and target to source), and the intersection $A = A1 \cap A2$ of the two sets of alignments are taken. Then for each English word *e* and Japanese word *j*, the frequency *N(e)* of *e* in *A* and the co-occurrence frequency

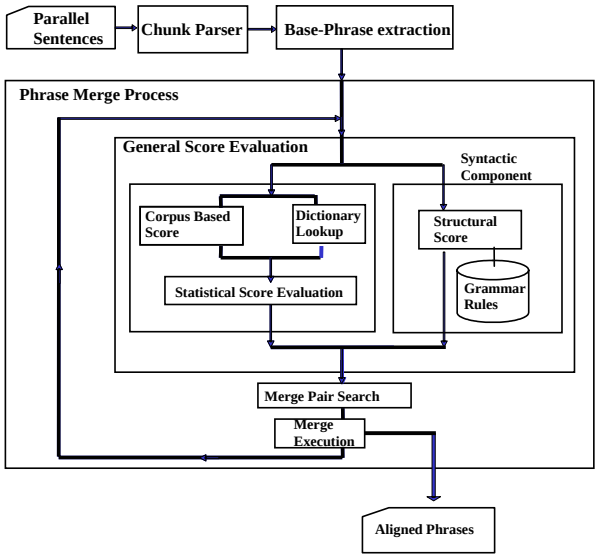


Figure 2: Framework of Phrase Aligner

N(e , j) of *e* and *j* in *A* are calculated. Furthermore, using a discrimination function $\delta(e, j)$ which determines whether *e* and *j* are a translation of each other with respect to a predefined bilingual dictionary, word based empirical translation probability is obtained as follows.

(2) $Pc(j|e) = (N(e,j) + \delta(e,j))/(N(e) + \Sigma_t \delta(e,t))$

$\delta(e, j)$ takes a value of 1 when *(e, j)* appears in the bilingual dictionary, and 0 otherwise.

An input to the phrase aligner is a pair **(J, E)** of Japanese and English sentences. The pair (**J**, **E**) is first chunk-parsed to extract base phrases, such as minimum noun phrases and phrasal verbs on both sides.

Let $\boldsymbol{J} = \boldsymbol{j}_1, \boldsymbol{j}_2, \ldots, \boldsymbol{j}_M$ be a series of Japanese chunks. These chunks are the minimum units for composing a final phrase alignment on Japanese side. Let $\mathbf{E} = w_1, w_2, \ldots, w_N$ be a series of English words. Then the probability that the translation of word w_i appears in chunk $\boldsymbol{j}_j$ in the given sentence pair is given by (3)[1].

(3) $P(\boldsymbol{j}_j | w_i) = C_{ij} / \Sigma_j C_{ij}$

, where

(4) $C_{ij} = \Sigma_t Pc(t|w_i)\, P(t \text{ appears in } \boldsymbol{j}_j)$

is what we will call *an alignment matrix* which represents the relative likelihood that the translation of word w_i appears in chunk $\boldsymbol{j}_j$ in comparison with other Japanese chunks, *t* is a translation candidate of w_i, and *P(t appears in* $\boldsymbol{j}_j$*)* is zero if $\boldsymbol{j}_j$ doesn't contain *t* as a substring and one if it does. Note that the values of C_{ij} can be calculated form the parallel sentence pair and the empirical translation probability (2).

Similarly for Japanese phrases, we can calculate the probability $P(w_i | \boldsymbol{j}_j)$ that the translation of $\boldsymbol{j}_j$ is represented as w_i as follows.

(5) $P(w_i \mid \boldsymbol{j}_j) = C_{ij} / \Sigma_i C_{ij}$

Given the translation probability (3), we can define the entropy $H(i)$ of the probability distribution $P(\cdot \mid w_i)$ as follows.

(6) $H(i) = -\Sigma_j P(\boldsymbol{j}_j \mid w_i) \log_2 P(\boldsymbol{j}_j \mid w_i)$

Since $\lim_{X \to 0} X \log_2 X = 0$, we define $H(i) = 0$ when $P(\boldsymbol{j}_j \mid w_i) = 0$ for all j.

In the proposed method, a statistical metric based on the entropy (6) is used for judging which adjacent phrases are to be merged. We calculate the change in the evaluation metric resulting from the merge just in the same way as we calculate the information gain (the reduction of entropy) of a decision tree when the dataset is divided according to some attribute, with the only difference that in a decision tree a dataset is incrementally *divided*, whereas in our method rows and columns are *merged*. We treat each row and each column of the alignment matrix as a dataset. The entire entropy, or uncertainty, of mapping English phrases to Japanese phrases is then given by:

(7) $H = \Sigma_i [\Sigma_j C_{ij}] H(i) / \Sigma_i \Sigma_j C_{ij}$

The entropy of mapping Japanese phrases to English phrases is obtained in the same way.

(8) $H_t = \Sigma_j [\Sigma_i C_{ij}] H(j) / \Sigma_i \Sigma_j C_{ij}$

Finally we define the total statistical metric, or an *evaluation score*, as the mean value of the two.

$H_{tot} = (H + H_t)/2$

Phrase Extraction

The merging process is terminated when the evaluation score H_{tot} takes a minimum value. When the final value of the alignment matrix is obtained, then for each non-zero

[1] Interested readers are referred to (Ushioda, 2007) for more details of the derivation of equation (3).

element C_{ij} the corresponding English phrase in the i-th row and the Japanese phrase in the j-th column are extracted and paired as an aligned phrase pair. Even after H_{tot} reaches zero we can continue merging as long as H_{tot} stays zero and a different set of phrase pairs can be extracted at each merging step while H_{tot} stays zero. Whether rows are merged or columns are merged at each merging step is determined by the evaluation score. Since the merging process is easily trapped by the local minimum with a greedy search, a beam search is employed while keeping multiple candidates (instances of alignment matrices). The typical beam size employed is between 300 and 1000.

One of the advantages of the proposed method is that we can directly incorporate dictionary information into the scheme, which is quite effective for alleviating data sparseness problem especially in the case of small training corpus. Another distinctive feature of the method is that once word alignments are obtained and the empirical translation probability *Pc(j|e)* is calculated together with the dictionary information, the word alignments are discarded. This is how this method avoids deterministic phrase alignment, and keeps a possibility of recovering from word alignment errors.

Multiple Correspondences

As we saw in the example of Figure 1 there is very often more than one element with a positive value in the same row of the alignment matrix. Usually only one nonzero element is correct and others are erroneously assigned nonzero values due to an accidental string match between the Japanese chunks and the translation of the English word. However there is no simple way of preliminarily disambiguating the identity of the corresponding Japanese chunk.

To cope with this initial ambiguity, a separate initial alignment matrix is constructed for each combination of a nonzero element of a row so that each row has at most one nonzero element. If there are *n* words $w_1, w_2, \cdots w_n$ in the English sentence, and each word w_i has k_i possible corresponding Japanese chunks, then the number of combinations is $k_1 k_2 \cdots k_n$, which sometimes becomes huge. However, in the process of merging, most of the erroneous word alignments disappear in confrontation with correct word alignments. Figure 3 shows two examples of an initial alignment matrix candidate for the sentence pair (1) and phrase alignments obtained after the merging process. Since the evaluation score of (c) is zero, (a) is considered to be the correct initial alignment matrix. As a result, the initial ambiguity on the identity of the corresponding Japanese chunk for each English word is resolved.

In some cases, however, multiple correspondences between English words and Japanese chunks are intrinsic. Consider the following sentence pair.

(11a) 真空賦勢した管及び血液の取り出し中に添加剤を分配するための方法を提供する。

(11b) *To provide a tube energized in vacuum and establish a method for distributing additives during the process of taking out the blood.*

Figure 4 shows the phrase alignment result for this pair and Figure 5 shows the initial and final alignment matrices. As Figure 4 shows the Japanese verb "提供する" (f) is aligned with both "To provide" (t) and "and establish" (v). This is because in the clausal conjunction different verbs are used for different objects (a tube and a method) in English whereas the same verb (f) is used in Japanese. In those cases one-to-one correspondence can never be achieved through merging, but still the evaluation score is expected to lead the merging process to a correct alignment result.

2.2 With Syntactic Information

The proposed framework also has a capability of incorporating syntactic constraints and preferences in the process of merging. For example, suppose that there are two competing merging candidates; one is to merge (i-th row, i+1-th row) and the other is to merge (k-th column, k+1-th column), and that their evaluation scores are H1 and H2 respectively. Then if there are no syntactic constraints or preferences, the merging candidate which has the lower evaluation score is elected. But if there are syntactic constraints, the only merging candidate which satisfies the constraints is executed. When a syntactic preference is introduced, then the evaluation score is multiplied by some value which represents the degree of the strength of the preference. If we intend to extract only pairs of phrases which constitute a constituent, then we introduce a constraint which eliminates merging candidates that produce a phrase which crosses a constituent boundary. Although our goal is to fully integrate complete set of CFG rules into the merging scheme, we are still in the process of constructing the syntactic rules, and in the present work we employed only a small set of preferences and constraints. Table 1 illustrates some of the syntactic constraints and preferences employed in the present work.

Merging lines or columns in the alignment matrix can be viewed as a form of bottom-up parsing. When we trace the process of the merging, its history can be converted to

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 To
0	0	0	0	0	0	0	0	0	0	0	0	0	64	1 calculate
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2 an
0	0	0	0	0	0	0	0	0	0	0	0	85	0	3 average
0	0	0	0	0	0	0	0	0	0	0	0	96	0	4 value
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5 weighed
0	0	0	0	0	0	0	0	0	0	0	0	0	0	6 in
0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	7 the
0	0	0	0	0	0	0	0	0	34	0	0	0	0	8 present
0	0	0	0	0	0	0	0	0	98	0	0	0	0	9 data
0	0	0	0	0	0	0	0	0	0	0	0	0	0	10 with
0	0	0	0	0	0	0	0	0	0	0	0	0	0	11 a
0	0	0	0	0	0	0	0	56	0	0	0	0	0	12 simple
0	0	0	0	0	0	0	0	97	0	0	0	0	0	13 arithmetic
0	0	0	0	0	0	0	0	99	0	0	0	0	0	14 circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	15 by
0	0	0	0	0	53	0	0	0	0	0	0	0	0	16 repeating
0	0	0	0	0	0	0	0	0	0	0	0	0	0	17 the
0	0	0	0	97	0	0	0	0	0	0	0	0	0	18 loop
0	0	0	0	0	0	0	0	0	0	0	0	0	0	19 of
0	72	0	0	0	0	0	0	0	0	0	0	0	0	20 multiplicatio
0	0	0	0	0	0	0	0	0	0	0	0	0	0	21 of
0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	22 the
27	0	0	0	0	0	0	0	0	0	0	0	0	0	23 stored
96	0	0	0	0	0	0	0	0	0	0	0	0	0	24 value
0	0	0	0	0	0	0	0	0	0	0	0	0	0	25 in
0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	26 the
97	0	0	0	0	0	0	0	0	0	0	0	0	0	27 arithmetic
99	0	0	0	0	0	0	0	0	0	0	0	0	0	28 circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	29 and
0	0	0	0	0	0	0	0	0	0	0	0	0	0	30 the
0	0	0	12	0	0	0	0	0	0	0	0	0	0	31 addition
0	0	0	0	0	0	0	0	0	0	0	0	0	0	32 of
0	0	0	0	0	0	0	0	0	0	0	0	0	0	33 a
0	0	20	0	0	0	0	0	0	0	0	0	0	0	34 new
0	0	98	0	0	0	0	0	0	0	0	0	0	0	35 data

(a)

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 To
0	0	0	0	0	0	0	0	9	0	0	0	0	0	1 calculate
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2 an
0	0	0	0	0	0	0	0	0	0	0	0	85	0	3 average
0	0	0	0	0	0	0	0	0	0	0	0	96	0	4 value
0	0	0	0	0	0	0	0	0	0	0	0	0	0	5 weighed
0	0	0	0	0	0	0	0	0	0	0	0	0	0	6 in
0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	7 the
0	0	0	0	0	0	0	0	0	34	0	0	0	0	8 present
0	0	0	0	0	0	0	0	0	98	0	0	0	0	9 data
0	0	0	0	0	0	0	0	0	0	0	0	0	0	10 with
0	0	0	0	0	0	0	0	0	0	0	0	0	0	11 a
0	0	0	0	0	0	0	0	56	0	0	0	0	0	12 simple
0	0	0	0	0	0	0	0	97	0	0	0	0	0	13 arithmetic
0	0	0	0	0	0	0	0	99	0	0	0	0	0	14 circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	15 by
0	0	0	0	0	53	0	0	0	0	0	0	0	0	16 repeating
0	0	0	0	0	0	0	0	0	0	0	0	0	0	17 the
0	0	0	0	97	0	0	0	0	0	0	0	0	0	18 loop
0	0	0	0	0	0	0	0	0	0	0	0	0	0	19 of
0	72	0	0	0	0	0	0	0	0	0	0	0	0	20 multiplication
0	0	0	0	0	0	0	0	0	0	0	0	0	0	21 of
0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	22 the
27	0	0	0	0	0	0	0	0	0	0	0	0	0	23 stored
96	0	0	0	0	0	0	0	0	0	0	0	0	0	24 value
0	0	0	0	0	0	0	0	0	0	0	0	0	0	25 in
0	0	0	0	0	0	0	0	0	0	0	0	0	0	26 the
0	0	0	0	0	0	0	0	97	0	0	0	0	0	27 arithmetic
0	0	0	0	0	0	0	0	99	0	0	0	0	0	28 circuit
0	0	0	0	0	0	0	0	0	0	0	0	0	0	29 and
0	0	0	0	0	0	0	0	0	0	0	0	0	0	30 the
0	0	0	12	0	0	0	0	0	0	0	0	0	0	31 addition
0	0	0	0	0	0	0	0	0	0	0	0	0	0	32 of
0	0	0	0	0	0	0	0	0	0	0	0	0	0	33 a
0	0	20	0	0	0	0	0	0	0	0	0	0	0	34 new
0	0	98	0	0	0	0	0	0	0	0	0	0	0	35 data

(b)

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]		
0	0	0	0	0	0	0	64	:	To calculate
0	0	0	0	0	0	182	0	:	an average value
0	0	0	0	0	133	0	0	:	weighed in the present data
0	0	0	0	252	0	0	0	:	with a simple arithmetic circuit
0	0	0	151	0	0	0	0	:	by repeating the loop
0	72	0	0	0	0	0	0	:	of multiplication
321	0	0	0	0	0	0	0	:	of the stored value in the arithmetic circuit
0	0	131	0	0	0	0	0	:	and the addition of a new data

[0]: 演算回路の記憶値の
[1]: 乗算と
[2]: 新しいデータの加算の
[3]: ループを繰り返すことにより,
[4]: 簡単な演算回路で
[5]: 現在のデータに重みを置いた
[6]: 平均値を
[7]: 算出可能とする

(c)

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]		
0	0	0	0	0	9	0	0	0	:	To calculate
0	0	0	0	0	0	0	182	0	:	an average value
0	0	0	0	0	0	133	0	0	:	weighed in the present data
0	0	0	0	0	252	0	0	0	:	with a simple arithmetic circuit
0	0	0	0	53	0	0	0	0	:	by repeating
0	0	0	97	0	0	0	0	0	:	the loop
0	72	0	0	0	0	0	0	0	:	of multiplication
124	0	0	0	0	0	0	0	0	:	of the stored value
0	0	0	0	0	196	0	0	0	:	in the arithmetic circuit
0	0	0	12	0	0	0	0	0	:	and the addition
0	0	119	0	0	0	0	0	0	:	of a new data

[0]:演算回路の記憶値の
[1]:乗算と
[2]:新しいデータの
[3]:加算のループを
[4]:繰り返すことにより,
[5]:簡単な演算回路で
[6]:現在のデータに重みを置いた
[7]:平均値を
[8]:算出可能とする

(d)

Figure 3: Two examples of an initial alignment matrix candidate for the sentence pair (1) and their merging results. (c) and (d) are the results of merging (a) and (b), respectively.

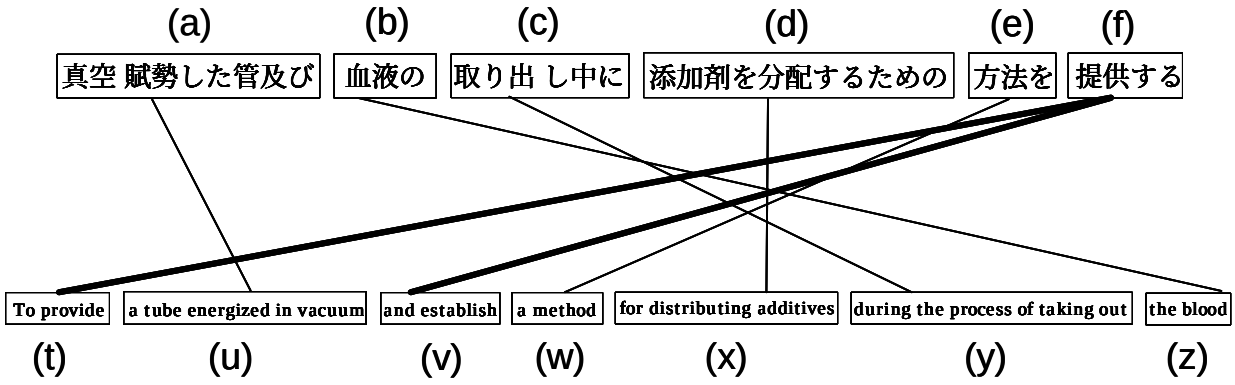


Figure 4: An example of intrinsic multiple correspondences.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	
0	0	0	0	0	0	0	0 To
0	0	0	0	0	0	83	1 provide
0	0	0	0	0	0	0	2 a
53	0	0	0	0	0	0	3 tube
0	0	0	0	0	0	0	4 energized
0	0	0	0	0	0	0	5 in
91	0	0	0	0	0	0	6 vacuum
0	0	0	0	0	0	0	7 and
0	0	0	0	0	0	23	8 establish
0	0	0	0	0	0	0	9 a
0	0	0	0	0	95	0	10 method
0	0	0	0	0	0	0	11 for
0	0	0	0	43	0	0	12 distributing
0	0	0	70	0	0	0	13 additives
0	0	0	0	0	0	0	14 during
0	0	0	0	0	0	0	15 the
0	0	1	0	0	0	0	16 process
0	0	0	0	0	0	0	17 of
0	0	68	0	0	0	0	18 taking
0	0	0	0	0	0	0	19 out
0	0	0	0	0	0	0	20 the
0	94	0	0	0	0	0	21 blood

[0]:真空賦勢した管及び
[1]:血液の
[2]:取り出し中に
[3]:添加剤を
[4]:分配するための
[5]:方法を
[6]:提供する

(a)

[0]	[1]	[2]	[3]	[4]	[5]		
0	0	0	0	0	83	:	To provide
144	0	0	0	0	0	:	a tube energized in vacuum
0	0	0	0	0	23	:	and establish
0	0	0	0	95	0	:	a method
0	0	0	113	0	0	:	for distributing additives
0	0	68	0	0	0	:	during the process of taking out
0	94	0	0	0	0	:	the blood

[0]:真空賦勢した管及び
[1]:血液の
[2]:取り出し中に
[3]:添加剤を分配するための
[4]:方法を
[5]:提供する

(b)

Figure 5: Initial (a) and final (b) alignment matrices for sentence (11)

a binary parse tree on both language sides. Since we are not yet incorporating grammar rules in our phrase alignment system, the merge history-induced inner-structures of the obtained bilingual phrases are not quite linguistically intuitive, although the obtained phrases themselves are intended to be linguistically motivated. However, even within the current setting, the obtained alignment matrix can be useful for guiding parsing process or correcting parse results via interplay between parsers of both sides through the alignment matrix. Figure 6 illustrates an example. If we suppose that the Japanese parse tree is more reliable than the English parse tree, then the alignment matrix can be used to convert Japanese tree structure into English one and to correct the PP-attachment error of the original English parse tree in which "*by forming*" is attached to "*to perform*" instead of the correct attachment site which is the conjunction of the preceding two clauses.

3 Experimets

This section describes experiments with the proposed phrase alignment method. For the evaluation of the obtained phrase alignments, two types of experiments are conducted. One is to evaluate the F-measure of the obtained phrase alignments with respect to a hand crafted golden standard. The second type is to measure the quality of phrase-based SMT which uses the obtained phrase pairs as a bilingual corpus. Each experiment is described in the following subsections. We used the test collection of a parallel patent corpus from the Patent Retrieval Task of the 3rd NTCIR Workshop (2002) for training word alignments. The corpus comprises of patent abstracts of Japan (1995-1999) and their English translation produced at Japan Patent Information Organization. We extracted 150 thousand sentence pairs from the PURPOSE part of the test collection of the year 1995. Each patent has its IPC category, from A through H. In-house English and Japanese parsers are used to chunk sentences and to make a constituent judgment. We also used in-house bilingual dictionary with 860 thousand word entries. For phrase alignment, we extracted 13,000 sentence pairs with English sentences of length smaller than 75 words, out of the sentence pairs in G-category (Physics) of the above word alignment training set. The sentence length is constrained to reduce the computational load. Table 2 summarizes the training corpora used. Out of 13,000 sentence pairs 208 thousand unique phrase pairs are extracted. More than one set of phrase alignments can often be extracted from one pair of aligned sentences when the evaluation score reaches zero.

Figure 7 shows examples of obtained phrase alignments. Japanese phrases acquired are mostly constituents, whereas many of English phrases are not, such as " by arranging", or "of infrared absorption ink". This is partly due to the fact that Japanese phrases are constructed out of base phrases, or chunks, whereas English phrases are constructed starting from individual words. Another reason is the fact that Japanese precedence rule takes precedence over English one as stated in Table 1.

3.1 Evaluation of Phrases with Human Judgment

Out of the 13,000 sentence pairs used for phrase alignments, 160 sentence pairs are randomly extracted for manual annotation. Although there have been a number of attempts to manually annotate word alignments, much less attempts have been made to construct a golden standard for phrase alignments. The major difficulty of aligning phrases is that there are many possible ways of aligning phrases, whereas word alignments have not much ambiguity.

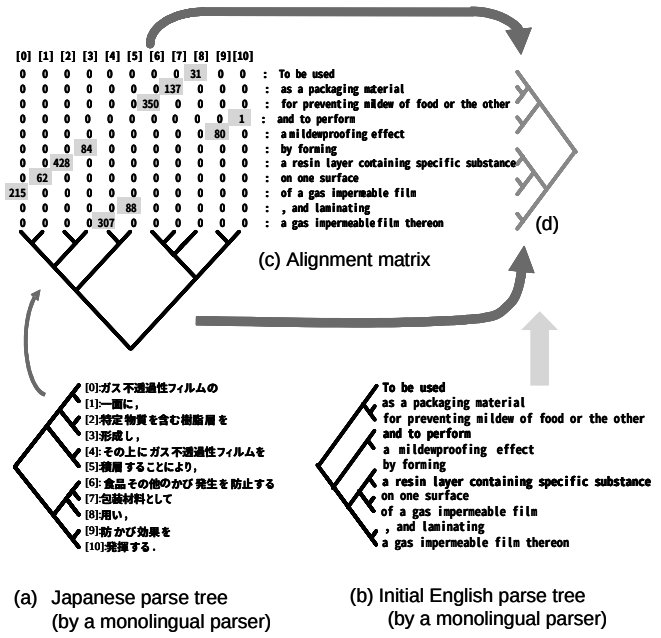


Figure 6: An example of correcting an English parse result by the combination of Japanese parse tree and the alignment matrix. In the initial English parse tree (b), the phrase "*by forming*" can be interpreted to be attached to "*and to perform*". Through the alignment matrix (c), the Japanese parse tree (a) can be automatically mapped to the English parse tree (d) which can for instance derive the correct interpretation of the attachment site of the phrase "*by forming*".

Since there is no obvious criterion to decide which phrase pairs are superior and which are not, we choose to extract all the possible ways of dividing a sentence pair into a set of bilingual phrases. Of course it is too much work for a human to exhaust all the possible combinations. However, there is a way of automatically generating all the possible phrase alignments from a result of manual work which is just repeating a simple task of dividing a phrase pair into pairs of sub-phrases. For example, consider a phrase pair in Figure 8. The phrase pair ("j1j2j3j4j5", "e1e2e3e4e5") is first divided into two phrase pairs, ("j1j2", "e4e5") and ("j3j4j5", "e1e2e3"). There are in total four possible division steps like this:

(12a) ("j1j2j3j4j5", "e1e2e3e4e5") ⇒ ("j1j2", "e4e5") , ("j3j4j5", "e1e2e3")
(12b) ("j1j2", "e4e5") ⇒("j1", "e5"), ("j2", "e4")
(12c) ("j3j4j5","e1e2e3") ⇒("j3","e3"), ("j4j5", "e1e2")
(12d) ("j4j5", "e1e2") ⇒ ("j4", "e2"), ("j5", "e1")

Given these four possible divisions, all the possible phrase alignments can be automatically calculated and the results are as follows.

("j1j2", "e4e5") ,("j3j4j5", "e1e2e3")
("j1j2", "e4e5"), ("j3","e3"), ("j4j5", "e1e2")
("j1j2","e4e5"),("j3","e3"),("j4","e2"), ("j5","e1")
("j1", "e5"), ("j2", "e4"), ("j3j4j5", "e1e2e3")
("j1","e5"),("j2","e4"),("j3","e3"),("j4j5", "e1e2")
("j1","e5"),("j2","e4"),("j3","e3"),("j4","e2"), ("j5","e1")

Therefore the task of human annotator is to keep dividing a phrase pair into pairs of sub-phrases. The procedure of the manual annotation is as follows.

1) Let the aligned sentence pair be a pair of aligned phrases.
2) Pick a pair of aligned phrases and try to divide it into two constituents so that each of the Japanese sub-phrases can be regarded as a translation of either of the English sub-phrases. An Example is given in Figure 9(a) and 9(b).
3) If 2) succeeds, repeat steps 2) through 4). If 2) fails, then try to divide the picked aligned pair of phrases into three, four, or more constituents in turn so that each of Japanese sub-phrases can be regarded as a translation of either of the English sub-phrases.
4) If 3) succeeds, repeat steps 2) through 4). Otherwise stop dividing the current pair of phrases and go through steps 2) through 4) with the next pair of phrases. If no more pair of phrases is available for dividing, terminate and output the set of division steps.

Figure 9 shows an example of dividing a pair of sentences into aligned phrases. The set {(a), (b)} constitutes one division step like (12a), as is also the case with sets {(c), (d)} and {(e), (f)}. From manually created division steps for the 160 sentence pairs, all the possible phrase alignments are generated and stored as a set of golden standard. Outputs of phrase aligner for these 160 sentences are then compared with the golden standard. For each phrase alignment in the golden standard, F-measure is calculated with the system output, and the maximum value among all the phrase alignments of the golden standard is recorded as the F-measure of the system output. The mean value of the F-measures of all the 160 sentences was 80.4. The average number of phrases in a sentence for the golden standard phrase alignments which give the maximum F-measure was 6.0. Therefore it is not the case that the most simple phrase alignment, which is a partition of a sentence into two parts, is earning high F-measures. In order to examine the contribution of simple phrase alignments, F-measures are calculated by gradually eliminating

	Constraint	Preference
Japanese	･conjunctions and punctuations are merged with the preceding entities	･when the score ties, a merge which creates a constituent takes precedence
English	･conjunctions, prepositions and punctuations are merged with the following entities ･merging across base-phrase boundary is pro-hibited	･when the score ties, a merge which creates a constituent takes precedence. If the English pref-erence conflicts with the Japanese prece-dence, the latter takes precedence.

Table 1: Syntactic constraints and preferences

Training	year	size(sent)	IPC CAT
Word Alignment	1995	150,000	A-H
Phrase Alignment	1995	13,000	G

Table 2: Training set description

from golden standard phrase alignments with small number of phrases. Table 3 shows the result. There are no big drops until MinNum = 4 , and after that F-measure declines rather rapidly. This also suggests that golden standard phrase alignments with 2 or three phrases are not playing a major role in the evaluation of the system outputs.

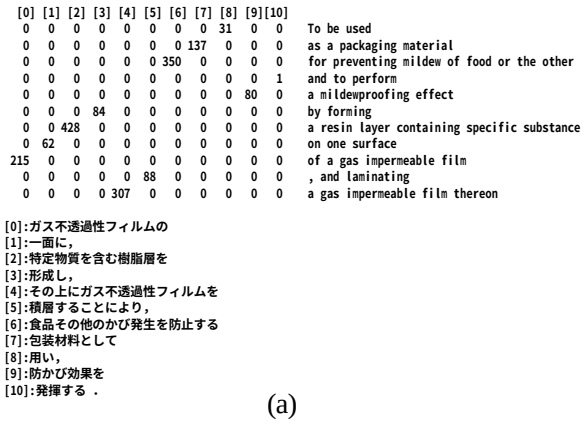
3.2 Evaluation of Phrases with SMT

The extracted phrase alignments were also evaluated with an SMT engine. We used Pharaoh (Koehn, 2004) as the baseline. Although our goal is to use obtained phrase alignments as translation units of Rule-based/SMT hybrid systems, we haven't yet processed large amount of parallel corpora, and the decoding scheme which takes advantage of the constituent oriented phrase alignments is still under development. Therefore, instead of testing the phrase alignments as translation units, we tested the cross-domain portability of the obtained phrase alignments. One of the major merits of a syntactic constituent is its generalization capability. N-gram statistics extracted from a large collection of data in a specific domain is a powerful resource within the same domain, but quite often fails to adopt to quite different domains. Constituents, or grammatical categories, on the other hand, cannot easily be tuned to a specific domain, but possess a generalization capability. In this experiment we trained Pharaoh using parallel sentences in one domain, namely IPC-G category (Physics), and tested the decoder in different domains. The training corpus we used for a baseline setting is the 13,000 sentence pairs in IPC-G category listed in Table 2 . We then used a set of aligned phrases extracted from the 13,000 sentence pairs for training Pharaoh (PhrAlign). The phrases are used alone and not mixed with the original parallel sentences. For testing, a set of 500 sentence pairs are randomly extracted from each IPC category. For development, another set of 500 sentence pairs are extracted from IPC-G category. Table 4 shows the result. PhrAlign outperforms Baseline in all the categoris. Especially in category E, PhrAlign scores 1.49 points higher than Baseline, which is relative percentage of 16% increase from Baseline. Since the training corpus is fairly small it is possible that the difference of the two cases decreases as the training data is increased, but this result suggests a generalizing capability of the syntactically oriented phrase alignments.

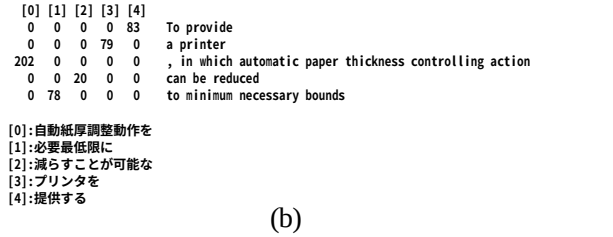
4 Related work

The inversion transduction grammar formalism (Wu, 1997) is one of the pioneering approaches for stochastically extracting bilingual phrases with constituent structure. A concept of bilingual parsing, where the input is a sentence pair rather than a sentence, is introduced in this framework. By allowing the inverse order of the right-hand-side of productions, the expressiveness of the grammar is shown to be considerably enhanced. In order to control the computational complexity, however, several severe constraints are applied, which makes it difficult to apply ITG to free-word-order languages like Japanese. This formalism is also not intended to be robust against the translation lexicon inadequacies: sentences containing more than one word absent form the translation lexicon are rejected in the reported experiment. The proposed method, on the other hand, is quite robust to a sparse alignment matrix because of the utilization of statistical word-alignment and the robustness of the chunkers.

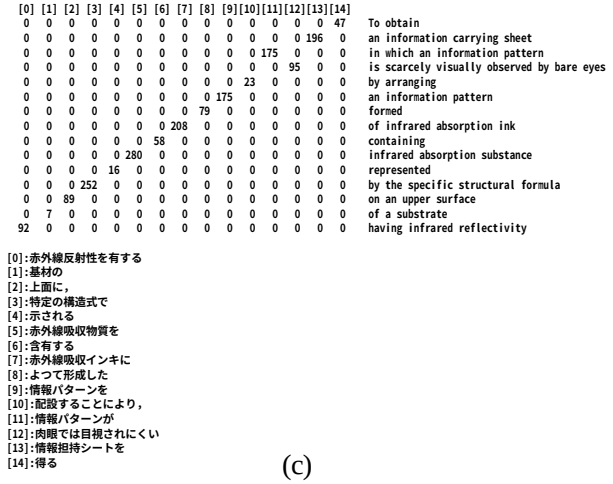
Integrated Segmentation and Alignment (Zhang and Vogel, 2005), or ISA, is probably most similar in concept to the proposed approach. ISA employs a greedy algorithm, called CGA, to extract phrase pairs out of a bilingual corpus. CGA extends the competitive linking algorithm (Melamed, 1997), a greedy word alignment algorithm with one word-to-one word assumption, to allow for combining



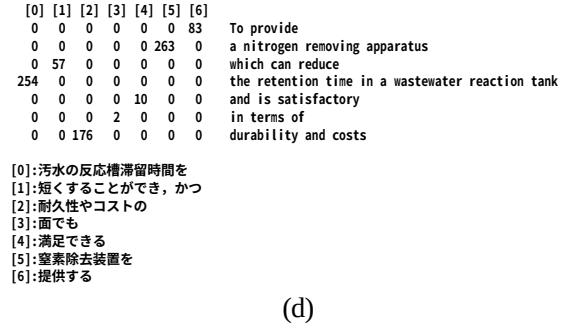
(a)



(b)



(c)



(d)

Figure 7: Examples of obtained phrase alignments

Min num of phrases	2	3	4	5	6	7
F-measure	80.4	78.4	78.4	72.6	69.6	64.6

Table 3: F-measure with minimum number of phrases in the golden standard varied

the detected "sure" word pair (a seed) with its neighbors to form a group. ISA uses χ^2 statistics to measure the mutual translation likelihood between words, and the word pair with the highest χ^2 value is selected as a seed. Neighboring words to be joined with the seed are also greedily searched on the basis of χ^2 values. Although both approaches use a statistical measure for the decision of agglomeration, CGS uses a word-to-word association for the judgment of local grouping, whereas the proposed approach uses a sentence level, or global, association metric for the judgment of merging, which makes the merging judgment justifiable not only for the merged phrase pairs, but also for the other words and phrases in the sentence pair. The n-best search in the proposed method also avoids the greediness of the merging process. Another difference is that in order to make the computation tractable, ISA employs a "locality assumption" which requires that a source phrase of adjacent words only be aligned to a target phrase composed of adjacent words. This assumption is again not suitable for language pairs of a quite different word order like the pair of Japanese and English.

5 Conclusion

A novel approach is presented for extracting syntactically motivated phrase alignments. In this method we

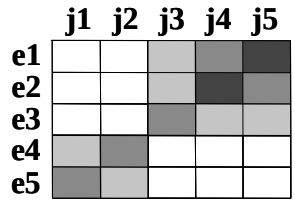


Figure 8: Multilayered Phrase Correspondences

IPC CAT	A	B	C	D	E	F	G	H
Baseline	7.94	11.43	10.24	7.42	9.29	11.38	14.66	12.03
PhrAlign	8.91	11.78	10.85	8.37	10.78	12.48	15.70	13.08

Table 4: Bleu score of the baseline and the proposed method.

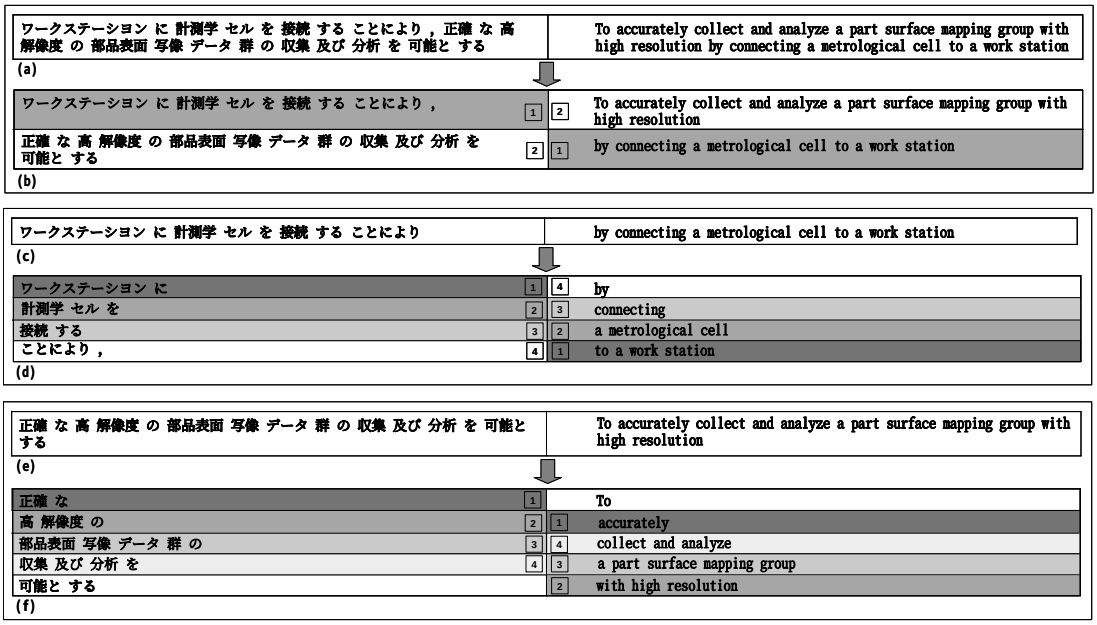


Figure 9: Manual Annotation of Phrase Alignments

can incorporate conventional resources such as dictionaries and grammar rules into a statistical optimization framework for phrase alignment. The method extracts bilingual phrases by incrementally merging adjacent words or phrases on both source and target language sides in accordance with a global statistical metric along with constraints and preferences composed by combining statistical information, dictionary information, and also grammatical rules. The extracted phrases achieved a maximum F-measure of over 80 with respect to human judged phrase alignments. The extracted phrases used as a training corpus for a phrase-based SMT showed better cross-domain portability over conventional SMT framework.

References

Chiang, David (2005). A hierarchical phrase-based model for statistical machine translation. In Proceedings of the 43rd Annual Meeting of the ACL (pp263-270).

Koehn, Philipp, Franz Josef Och,, and Daniel Marcu (2003). Statistical Phrase-Based Translation. In Proceedings of HLT-NAACL.

Koehn, Philipp (2004). Pharaoh: a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In 6th Conference of the Association for Machine Translation in the Americas, AMTA.

Melamed, I. Dan (1997). A Word-to-Word Model of Translational Equivalence. In Proceedings of the Eighth Conference of the European Chapter of the Association for Computational Linguistics (pp.490-497).

Marcu, Daniel and William Wong (2002). A Phrase-based Joint Probability Model for Statistical Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp.133-139).

NTCIR Workshop (2002). http://research.nii.ac.jp/ntcir/ntcir-ws3/work-en.html.

Och , Franz Josef and Hermann Ney (2000). Improved statistical alignment models. In Proceedings of the 38th Annual Meeting of the ACL (pp.440-447).

Och, Franz-Josef and Hermann Ney (2004). The alignment template approach to statistical machine translation. Computational Linguistics, 30(4), 417--450.

Ushioda, Akira (2007). Phrase Alignment for Integration of SMT and RBMT Resources. In Proceedings of MT Summit XI Workshop on Patent Translation (to appear), Copenhagen, Denmark.

Wu, Dekai (1997). Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. Computational Linguistics, 23(3), 377-403.

Yamada , Kenji and Kevin Knight (2001). A syntax-based statistical translation model. In Proceedings of the 39th Annual Meeting of the ACL (pp.523-530).

Zhang, Ying and Stephan Vogel (2005). Competitive Grouping in Integrated Phrase Segmentation and Alignment Model. In Proceedings of the ACL Workshop on Building and Using Parallel Texts (pp 159—162).

TMI-07

11th International Conference on Theoretical and Methodological Issues in Machine Translation

List of Authors