XIII ANNUAL CONFERENCE

EAMT

14-15 May 2009 Barcelona, Spain

Universitat Politècnica de Catalunya

Proceedings of the 13th Annual Conference of the European Association for Machine Translation

14–15 May 2009

Universitat Politècnica de Catalunya

Barcelona, Spain

EUROPEAN ASSOCIATION FOR MACHINE TRANSLATION

UPC

TALP

Published by
Universitat Politècnica de Catalunya
C. Jordi Girona, 31
08034 Barcelona
Spain

ISBN:

Printed by
SERVIS GRÀFICS COPISTERIA IMATGE, S.L.
Cinca, 8
08030 Barcelona
Spain

Conference Sponsors

Order copies of this and other EAMT proceedings from:

EAMT Secretariat
c/o A. Clarke
Schützenweg 57
CH-4123 Allschwil
Switzerland
secretariat@eamt.org
http://www.eamt.org/

Foreword

The European Association for Machine Translation (EAMT) has a long tradition for organising annual workshops and conferences with the purpose of bringing together people and companies with a professional interest in machine translation and other tools for translation, be it users, researchers, developers, or providers who want to follow latest developments.

The term "machine translation", MT, is interpreted in its widest sense at the EAMT conferences. This means that MT refers not only to fully automatic translation, but also to all other kinds of tools for translation or multilinguality, be it translation memory, parallel corpora and other resources for translations, alignment, terminology tools, etc.

This year, the focus of the EAMT conference is on how to develop translation technologies for and among languages having smaller speech communities or limited digital resources. Actually most of the world's languages are of this category, and we are sure that researchers, users and providers have an interest in such technologies.

The fact that researchers and users with explicit needs are brought together, and that there is a fair number of research presentations as well as some more practically oriented presentations, provide excellent opportunities for mutual feedback. Through our annual conferences we have been able to create an environment for interesting discussions and maybe even for the creation of new partnerships, and we are sure that this conference will contribute to the continued success of the EAMT conferences.

EAMT is deeply dependent on colleagues willing to take upon themselves the tasks of programme committee work and local organization.

So, before closing, I first want to thank all of the programme committee for their invaluable contribution, not least the programme committee co-chairs Lluís Màrquez and Harold Somers.

Secondly, I want to thank David Farwell, Adrián R. Fonollosa, José Mariño and other colleagues at the Centre for Speech and Language Applications and Technologies (TALP) at the Universitat Politècnica de Catalunya for the local organization. The MT research group at TALP kindly invited the EAMT to have the conference in Barcelona, and it has been a great pleasure for the EAMT Executive Committee to collaborate with them.

Finally: I wish all of you an excellent and enjoyable conference!

Bente Maegaard
Center for Sprogteknologi,
University of Copenhagen, Denmark
President of the EAMT

Message from the Programme Committee Chairs

We are delighted to welcome you to the 13th Annual Conference of the European Association for Machine Translation in Barcelona. Starting out as occasional workshops, I hope you will agree that the EAMT Annual Conference has now established itself in the calendar of important events in the field of MT.

As may be appropriate these days in our field, let us start with some statistics: we received 54 submissions, of which 14 were accepted as full papers, 18 as posters, providing a healthy overall acceptance rate of 59%. Submissions came from 24 different countries, including 9 beyond Europe's borders. The two countries providing most papers were Spain (10) and Ireland (8).

The decision to designate submissions as papers or posters was taken purely as a reflection of the most suitable form of presentation of the work, as determined by an explicit request to reviewers. While there are still some who think of posters as "second-class" papers, we would like to assure all presenters that this is far from the case. For a start, all papers are afforded equal space and prominence in these Proceedings. Furthermore, as more and more people are discovering, presenting one's work as a poster can have serious benefits and advantages over an oral presentation, not least of which are the possibility of engaging on a personal level with a select audience which has explicitly chosen to seek out your presentation and to participate in it. More than a few presenters are now coming to the idea that a poster can actually be more effective than the traditional but impersonal lecture style of presentation.

As programme chairs, we are of course indebted to the panel of reviewers, whose names are listed elsewhere. 45 reviewers looked at 3 or 4 papers each, thus ensuring that each submission got three separate reviews. We asked reviewers to work to a tight schedule, and almost without exception they got their reviews in on time, which in turn meant that we could notify authors of acceptance or, regrettably, rejection, even a few days before our stated deadline. We hope that authors – whether chosen or not – have appreciated and benefited from the reviewers' comments, which were often quite extensive. Equally we are grateful to authors who were asked to prepare their final copy for these Proceedings within a fairly short deadline. Again, with only one or two exceptions, the deadline was met, and we were able to avoid the usual panic and scramble associated with this task.

As Programme chairs, our job sort of ends once we have chosen which papers to accept, and arranged them into the programme you will experience and, we hope, enjoy over the next two days. At this point we hand over to the Local Organisation Committee, but of course we have been working closely together with them since day one, a task obviously much facilitated by the fact that one of us is a "local". Nevertheless, the local organisers have been with us every step of the way, and we would like here to thank them for their support, advice and, when necessary, gentle prodding. Of great help too have been the EAMT Executive Committee, with advice on precedent, form and character, so that this conference should at the same time fit comfortably in with the EAMT conference series, meeting the needs and expectations of members of the EAMT, while, we hope, standing out as a memorable and enjoyably different conference.

Finally, thanks to all authors, presenters and attendees for making this a successful 13th Annual EAMT Conference.

Lluís Màrquez and **Harold Somers**
EAMT–2009 Programme Committee co-chairs

Message from the Local Organising Committee

It gives us great pleasure to welcome you to the EAMT 2009, the 13th Annual Conference of the European Association for Machine Translation. This year the conference is being held on the Campus Nord of the Universitat Politècnica de Catalunya in Bacrelona, Spain. We have tried to make all the necessary arrangements to ensure that your participation in the conference events is as productive and enjoyable as possible. While in Barcelona be sure to experience the special atmosphere the city has to offer: the Roman, medieval, and modernist architecture of the old city, Passeig de Gràcia and the Eixample and the wide array of excellent restaurants, theatre, music, galleries and museums. It would be unfortunate not to take in all you can while here.

Of course, we also hope that you will benefit from a strong programme of conference presentations and workshops which are at the forefront of MT and multilingual language processing research and development.

In organising the conference we have received significant financial support from the Universitat Politècnica de Catalunya and the Spanish Ministry for Science and Innovation. We also would like to thank European Languages Resources Association (ELRA) and Springer for their generous sponsorship. Finally, we have also had the unselfish assistance of local staff and students. In particular we wish to thank Coralí Planellas for her many hours of effort, especially in maintaining the conference web site.

So without further ado, welcome and enjoy the conference.

Local Organising Committee:

David Farwell
José A. R. Fonollosa
José Mariño
Lluís Màrquez

Centre de Tecnologies i Aplicacions del Llenguatge i la Parla
Universitat Politècnica de Catalunya

EAMT-2009 Organizers

Programme Committee Chairs:

Lluís Màrquez, Universitat Politècnica de Catalunya, Spain
Harold Somers, Dublin City University, Ireland

Program Committee:

Joseba Abaitua, Universidad de Deusto, Spain
Iñaki Alegria, Euskal Herriko Unibertsitatea, Spain
Juan Alonso, Translendium SL, Spain
Toni Badia, Universitat Pompeu Fabra, Spain
Rafael Banchs, Barcelona Media Innovation Centre, Spain
Pierette Bouillon, Université de Genève, Switzerland
Chris Callison-Burch, Johns Hopkins University, USA
Nicola Cancedda, Xerox Research Centre Europe, France
Michael Carl, Copenhagen Business School Handelshøjskolen, Denmark
Francisco Casacuberta, Universitat Politècnica de València, Spain
Irene Castellón, Universitat de Barcelona, Spain
Adrià de Gispert, Cambridge University, England
Arantza Díaz de Ilarraza, Euskal Herriko Unibertsitatea, Spain
Bonnie Dorr, University of Maryland, USA
Andreas Eisele, DFKI GmbH, Germany
David Farwell, Universitat Politècnica de Catalunya, Spain
Marcello Federico, Fondazione Bruno Kessler, Italy
José A. R. Fonollosa, Universitat Politècnica de Catalunya, Spain
Mikel Forcada, Universitat d'Alacant, Spain
Jesús Giménez, Universitat Politècnica de Catalunya, Spain
John Hutchins, United Kingdom
Kevin Knight, University of Southern California, USA
Philipp Koehn, University of Edinburgh, Scotland
Alon Lavie, Carnegie Mellon University, USA
Lori Levin, Carnegie Mellon University, USA
Bente Maegaard, Københavns Universitet, Denmark
Daniel Marcu, University of Southern California, USA
José B. Mariño, Universitat Politècnica de Catalunya, Spain
M. Antònia Martí, Universitat de Barcelona, Spain
Hermann Ney, Aachen University, Germany
Sharon O'Brien, Dublin City University, Ireland
Stephan Oepen, University of Oslo, Norway
Mike Rosner, University of Malta, Malta
Marta R. Costa-jussà, Universitat Politècnica de Catalunya, Spain
Anna Sågvall Hein, Uppsala universitet, Sweden
Kepa Sarasola, Euskal Herriko Unibertsitatea, Spain
Holger Schwenk, Université du Maine, France
Hisami Suzuki, Microsoft Research, USA
Cristina Vertan, Universität Hamburg, Germany
Walther von Hahn, Universität Hamburg, Germany
Andy Way, Dublin City University, Ireland
Dekai Wu, The Hong Kong University of Science and Technology, Hong Kong

Additional Reviewers:

Mauro Cettolo, Marine Carpuat, Maja Popović, Nicola Bertoldi, Felipe Sánchez-Martínez, Gonzalo Iglesias, Saab Mansour, Elisabeth Comelles

Invited Speakers:

Lori Levin, Carnegie Mellon University, USA
Nicholas Ostler, Foundation for Endangered Languages

Local Organising Committee:

David Farwell
José A. R. Fonollosa
José Mariño
Lluís Màrquez

Centre de Tecnologies i Aplicacions del Llenguatge i la Parla
Universitat Politècnica de Catalunya

Table of Contents

Conference Program

Thursday, May 14, 2009

9:00–9:15 Welcome

9:15–10:15 Invited Talk
"The Jungle is Neutral" – Newcomer Languages Face New Media
Nicholas Ostler

10:15–10:45 *Character-Based PSMT for Closely Related Languages*
Jörg Tiedemann

10:45–11:15 *TS3: an Improved Version of the Bilingual Concordancer TransSearch*
Stéphane Huet, Julien Bourdaillet and Philippe Langlais

11:15–11:45 Coffee break

11:45–12:15 *Estimating the Sentence-Level Quality of Machine Translation Systems*
Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman and Nello Cristianini

12:15–12:45 *Evaluation-Guided Pre-Editing of Source Text: Improving MT-Tractability of Light Verb Constructions*
Bogdan Babych, Anthony Hartley and Serge Sharoff

12:45–13:15 *Learning Labelled Dependencies in Machine Translation Evaluation*
Yifan He and Andy Way

13:15–14:30 Lunch

14:30–15:00 *Improving a Catalan-Spanish Statistical Translation System using Morphosyntactic Knowledge*
Mireia Farrús, Marta R. Costa-jussà, Marc Poch, Adolfo Hernández and José B. Mariño

15:00–15:30 *Use of Rich Linguistic Information to Translate Prepositions and Grammar Cases to Basque*
Eneko Agirre, Aitziber Atutxa, Gorka Labaka, Mikel Lersundi, Aingeru Mayor and Kepa Sarasola

Thursday, May 14, 2009 (continued)

15:30–17:00 **Poster Session I**

Gappy Translation Units under Left-to-Right SMT Decoding
Josep M. Crego and François Yvon

Relevance of Different Segmentation Options on Spanish-Basque SMT
Arantza Díaz de Ilarraza, Gorka Labaka and Kepa Sarasola

English-Latvian Toponym Processing: Translation Strategies and Linguistic Patterns
Tatiana Gornostay and Inguna Skadina

An Environment for Named Entity Recognition and Translation
Filip Graliński, Krzysztof Jassem and MichałMarcińczuk

Optimal Bilingual Data for French–English PB-SMT
Sylwia Ozdowska and Andy Way

Word- and Sentence-Level Confidence Measures for Machine Translation
Sylvain Raybaud, Caroline Lavecchia, David Langlois and Kamel Smaïli

Translating Questions for Cross-Lingual QA
Jörg Tiedemann

Developing Prototypes for Machine Translation between Two Sámi Languages
Francis M. Tyers, Linda Wiechetek and Trond Trosterud

Collocations in a Rule-Based MT System: A Case Study Evaluation of their Translation Adequacy
Eric Wehrli, Violeta Seretan, Luka Nerima and Lorenza Russo

16:30–17:00 Coffee break

17:00–17:30 *Automatic Translation of Norwegian Noun Compounds*
Lars Bungum and Stephan Oepen

17:30–18:00 *Marker-Based Filtering of Bilingual Phrase Pairs for SMT*
Felipe Sánchez-Martínez and Andy Way

Friday, May 15, 2009

9:00–9:30 *Tree-Based Target Language Modeling*
Vincent Vandeghinste

9:30–10:00 *Language Model Adaptation for Difficult to Translate Phrases*
Behrang Mohit, Frank Liberato and Rebecca Hwa

10:00–11:00 Invited Talk
Adaptable, Community-Controlled, Language Technologies for Language Maintenance
Lori Levin

11:00–11:30 Coffee break

11:00–12:30 **Poster Session II**

A Phrase-Based Hidden Semi-Markov Approach to Machine Translation
Jesús Andrés-Ferrer and Alfons Juan

Building Strong Multilingual Aligned Corpora
Reza Bosagh Zadeh

A Constraint Satisfaction Approach to Machine Translation
Sander Canisius and Antal van den Bosch

Introducing the Autshumato Integrated Translation Environment
Hendrik J. Groenewald and Wildrich Fourie

A New Subtree-Transfer Approach to Syntax-Based Reordering for Statistical Machine Translation
Maxim Khalilov, José A. R. Fonollosa and Mark Dras

On Extracting Multiword NP Terminology for MT
Svetlana Sheremetyeva

Rule-Based Augmentation of Training Data in Breton-French Statistical Machine Translation
Francis M. Tyers

Can Semantic Role Labeling Improve SMT?
Dekai Wu and Pascale Fung

Are Unaligned Words Important for Machine Translation?
Yuqi Zhang, Evgeny Matusov and Hermann Ney

Friday, May 15, 2009 (continued)

12:30–13:30 EAMT Business Meeting

13:30–14:45 Lunch

14:45–15:15 *Using Supertags as Source Language Context in SMT*
Rejwanul Haque, Sudip Kumar Naskar, Yanjun Ma and Andy Way

15:15–15:45 *On LM Heuristics for the Cube Growing Algorithm*
David Vilar and Hermann Ney

15:45–16:15 Coffee break

16:15–16:45 *Tuning Syntactically Enhanced Word Alignment for Statistical Machine Translation*
Yanjun Ma, Patrik Lambert and Andy Way

16:45–17:00 Closing session – presentation of the Springer Award for best paper

"The Jungle Is Neutral" – Newcomer Languages Face New Media

Nicholas Ostler
Foundation for Endangered Languages
172 Bailbrook Lane
Bath BA1 7AA
England
nostler@chibcha.demon.co.uk

Abstract

The origin and early history of research and development in machine translation might suggest that it is only interesting, or applicable, to the greatest of major languages. But founders' effects do not persist in eras of furious technical change, unless they concern essentially arbitrary aspects, such as technical standards. Machine translation, and language technologies more generally, may yet be very useful to minority languages, promoting and extending both their use and their status, in a world where there may be more than one dominant language.

The title quote is borrowed from F. Spencer Chapman's 1949 book on his experience of jungle warfare.

1 Introduction

This is a time of great political and economic turmoil, when the future power-structures of the world are unknown, clouded by changes that have not as yet run their course. One of these unknown future structures is the space that will be available to the world's languages. Will the world's rising powers such China, India, Russia and Brazil simply accept the current dispensation, and communicate in the relatively neutral medium of English, alien and costly though it may be to many of them? But then, have they any power to set up an alternative? Will the world's minority languages continue to yield ground to the dominant languages in their various countries, losing functions, and more and more failing to be picked up at all by rising generations? Or will the emerging situation, technological as well as social and political, offer them new options for survival and utility? In fact, study of the past history of synergies between languages and language technologies suggests some surprising possibilities for the future.

The modern position of the English language clearly owes much to a generalized technological revolution, the massive increase in immediate wealth and power that came from the use of mass production, fossil-fuel-burning industry, and world-shrinking transport and information-exchange; this revolution came about just as the United Kingdom, and then the United States of America, were spreading their political power, and their agents of enterprise, to every corner of the world. These, the switch to the technologies and the spread of the language, happened principally from the 18th to the 20th centuries.

Much more specifically, the application of computer technology to machine translation (MT), as it happens, owes its first surge of development to competition between the USA and the Soviet Union of the 1950s-60s, taking in the early Cold War and the Space Race. It was then adopted and extended by other significant scientific and technical powers of those days, seen for a time (the 1980s and early 1990s) as an important enabling technology by governments such as those of Japan and the European Union. As the principles of the technology became better understood, there even began a drive to create a meta-system which might generate new MT systems on the fly, as and when English-speakers needed access to any other "low-density language ".

This history might in itself suggest that MT is only interesting for, or in practice applicable to, the greatest of major languages – Russian, Japanese, the state languages of western Europe, Chi-

Proceedings of the 13th Annual Conference of the EAMT, pages 1–7,
Barcelona, May 2009

nese and above all to English. And these languages have certainly carried off the early laurels for research and exploration, and limited success, in the field.

But founder effects – defined as the continuing dominance of those who have pioneered a move, even as others join in, and make their own contributions to it – do not necessarily persist in eras of furious technical or political change. For example, English has indeed continued as the language of foreign colonists who came to dominate North America, even though the native tongues of the vast majority of later immigrants were different, mostly Slavic or Germanic. But Portuguese has not sustained its early (16th to 18th century) role as the lingua franca of trade and diplomacy round the Indian Ocean. The collapse of Portuguese mercantile control in the 17th century did not benefit linguistically the Dutch who subverted it, but it did leave the field clear for the later growth of English.

There is no simple rule, then, that decrees the long-term triumph of those who first take up a technical option, or a dominant position in a new world-order. This possibility, of latecomer dominance, is as applicable to the mastery and use of machine-translation technology as it is to the general survival and role of individual languages within the world system. If the two applications are put together, it may even be that the late-coming use of MT technology to give access to other languages, for large or small communities, will give new life-chances to such languages.

To explore these prospects, it is reasonable to ask, and try to answer, three key questions. Firstly, what does it take to unseat an established lingua franca, inhibiting its continuing transmission down the generations? Secondly, can the wider application of MT provide what is required to do this? Thirdly, is the future course of globalization likely to call for some lingua franca in the foreseeable future, whether it will be English, or some successor to it?

2 To Unseat a Lingua franca

The most interesting case of an established lingua franca which came unstuck is that of Latin in Europe after the Renaissance. It is interesting in that there was no evident competitor which supplanted it, and many aspects of the situation, as well as contemporary events, might have been expected to support its role, and indeed to enhance it.

In the late 15th century, the printing press became available in Europe, first in Germany and then in Italy. The first book to be printed was, unsurprisingly, the Vulgate Bible in Latin, and to start with, the vast majority of books that came off the presses, wherever they might be in Europe, were in Latin. The whole of Latin literature that had survived from the classical era was soon in print, and now that texts could be duplicated without error through mechanical production, textual criticism could become systematic, best editions could be reconstructed through comparing the various available manuscripts, and the results agreed as standard. In addition, with books becoming cheap, scholars might now be expected to purchase their own copies, and classes could learn from textbooks. Many of the early best sellers were indeed Latin textbooks. Furthermore, since users of Latin made up the only language-community that could be assumed to be 100% literate (always having learnt the language at school), and since they were distributed throughout Europe except in the orthodox zone (of east and south-east), there would have seemed no doubt at all that Latin would offer the best language in which to print books, and to produce them in the longest print-runs.

At just this time, European mariners – first Portuguese and Spanish, then French, Dutch, English and Danish – discovered the vast potential of the world beyond Europe's coastal waters. European settlements sprang up in the illiterate and sparsely populated lands of the Americas, as well as in the high developed markets of the Indian Ocean, and China beyond. All the captains of these expeditions, as educated men, knew Latin, and the first accounts of their discoveries, by such writers and Peter Martyr, were circulated round Europe in that language. Although the different European nations were in competition to set up these settlements and trading posts, the Catholics among them were explicitly charged by their Pope to win souls for Christ on these expeditions, something that would have been unthinkable for them without use of the Latin language, especially if a native priesthood was to be established and educated in the new lands.

Nevertheless, just when Latin – the textually-based language par excellence – had finally got its classical texts firmly defined and economi-

cally distributed, and seemed poised to travel with European venturers as they spread their interests, their faith and soon their control, round the wider world, it began to lose its pre-emptive dominance. Educated discourse began to be acceptable in vernacular languages, first in the leading powers of the west, France and then England, then in the powers of central Europe, such as the Netherlands, Germany and Italy, and finally in the peripheral powers of the East and North, such as Austria, Hungary, Poland and Sweden. The book markets that had sprung up all over the continent switched during 16th and 17th centuries from Latin to the various vernaculars: even in the New World, the printing-presses were producing texts in the indigenous languages, which the Spanish missionaries had just succeeded in analysing and reducing to (roman) script.

Latin failed for a variety of reasons. One reason may have been "insufficient globalization" of the market: excessive costs of book transportation round Europe, as against the costs of book production, which meant that publishers and booksellers stood to gain more from selling their print-runs close to the home, with local audiences who read in the vernacular, than to an elite, pan-European, market, who could read in Latin. Close at hand, there were always more vernacular readers than Latinists, and now that books were produced in quantity rather than one at a time, those numbers began to count. But in addition to economics, there was a power-shift going on between the classes. The elite were less and less traditionally educated clerics, and more and more urban bourgeoisie who had had a more practical, and vernacular education. National governments too, led off by France and England, as they distanced themselves from Church power, wished to discriminate in favour of their own vernaculars. Already in 1539, King François I had required by the Ordinance of Villers-Cotterêts that official documents, whether from courts or parish registers, should all be produced *en langage maternel françois et non autrement* – "in French mother tongue, and not otherwise", implicitly not in Latin.

What, then, had inhibited the transmission of Latin as a lingua franca? No single language had stepped in to take its place, but the power-structure of society had changed. As people educated without Latin came to assume greater influence (and the influence of the greatest Latin-using power, the Roman Catholic Church, declined) there was simply less call for skill in Latin. The international contacts which were facilitated by use of Latin as a common language were naturally diminished. But this was less crucial, in a new society where separate nation-states dominated in their own interests. The "founder effect", that had transmitted Latin through a good millennium, or 40 generations, when it was not close to the vernacular for much of Europe's population, had been undone.

Founder effects, a.k.a. the force of tradition, are stronger where either there is little cost in sustaining the past pattern (contrast the continuing expense in time and effort to induct new generations in Latin, effectively an artificial language), or the tradition is not at variance with some other new pressure (as Latin was in effect a barrier to entry for less educated bourgeois people). Hence notoriously, wheel gauges have been sustained at 4 ft 8½ in from the Roman empire and its road ruts to the US standard railroad gauge. What motive was there to change as one style of wheeled transport succeeded another? One could also note that Renaissance typographers of the 15th and 16th centuries, choosing the character styles for printing fonts simply took over the styles (**Gothic**, Roman and *Italic*) which were then in vogue in manuscript hands. They have been sustained ever since – though Gothic, the least readable, has lost much ground – since there is no more of the particular dynamic in manual pen movement which had previously driven the changes since the CAPITALS of the Classical age. Even more notoriously, the perverse QWERTY pattern of the English keyboard, invented in 1872, has survived a century of mechanical typing and the first 30 years or so of digital text entry. It is likely to continue to survive unless and until it comes to represent a barrier to entry to some sector of the population of would-be typists and writers, which – hitherto disenfranchised – is yet rising in influence.

This kind of situation is precisely what can be expected to provide at least opposition, and perhaps effective revolution, to the retention of English as a global lingua franca. What about the vast section of the world's population for whom the need to learn English is still a burdensome chore which they would prefer to avoid?

3 When Founder Effects Live On

As a digression, or an examination of the clinging power of a dead or dying lingua franca, we may note that features of an inherited system are not always rationalized away. If they are harmless, or in some way emphasize the (conservative) power of a favoured group, they may be preserved. Hence in the cuneiform ideographic writing which was invented to write Sumerian, but subsequently adopted to write Akkadian, the rebus principle is operated to give punning meanings to characters, using both these languages. But when the script was later adopted to write Elamite, Hittite and Ugaritic, the alien puns were retained (as 'Sumerograms' and 'Akkadograms') in the writing system, although the pronunciation in the new languages would have followed the meaning rather than any attempt to borrow the Sumerian or Akkadian words literally.

This principle, understandable in ideographic scripts like cuneiform or Chinese characters, actually continued to be followed after Aramaic and then Persian came to be written with purely alphabetic scripts. Since, pragmatically, texts written in Aramaic to Persian addressees were usually read out only in Persian translation, the Aramaic came to be seen as an indirect way of writing the Persian. When later, Persian itself came to be written in the Aramaic alphabet, it would be interspersed with large numbers of words written alphabetically in Aramaic, but each to be pronounced as the synonymous word in Persian. These so-called 'Aramaeograms' persisted in use long after Aramaic itself had been forgotten by most scribes.

In certain cases, the pragmatics of these alien carry-overs may be not burdensome or neutral, but even beneficial to receiving users. This has been the case with the alphabetical input of Chinese characters in Chinese and Japanese word-processing.

It had long been an unsolved problem in attempting directly to input Chinese characters from a keyboard that there were just too many keys; when an operator had to chose from an array of several thousand characters, location just took too long, so that direct input by pen was usually preferred. However, in an electronic context, language technologists at Toshiba discovered that candidate characters could be located much more efficiently – and at a speed acceptably close to real-time – if their phonetics were typed in alphabetically, and the reduced set of candidate characters was then used to select the actual character desired. Furthermore, the phonetics are more efficiently typed in using the Roman alphabet than the traditional Japanese phonetic syllabary, the *kana*: such Roman phonetics never appear in the resulting text, but they do facilitate the entry of characters, whether in Japanese or in Chinese.

So in fact, an arbitrary carry-over of apparently irrelevant technical details from one language's technology to another system may, by good luck or good judgment, in fact solve that other's persistent problem. This happens because deep learning, as well as loss of traditional skill, can result from the introduction and acceptance of new technologies from alien sources.

4 Is MT Equal to the Task?

This all provides some kind of answer to the first question: "what does it take to unseat an established lingua franca?" In essence, the answer is the context needs to change so that what was an advantage comes to be seen as a net liability. We proceed to the second question: "can the wider application of Machine Translation provide what is required to do this?" Can the availability of MT cause such a change to the surrounding context for international communication that continuing use of English will be undercut?

Prima facie, the answer to this is unpromising. It has been an unchanging truism of MT, almost from the beginning of its fifty-year history, that its results have been disappointing. The hope that inspired, and for a long time funded, MT was that it could provide a cheap, fast and high-quality substitute for human translators or interpreters, so that in effect the language barrier would go away. This has not happened, though the reasons for this disappointment are not clear and distinct.

Ironically, this ambiguity was dramatized most memorably for me by the Danzin report, which in 1990 evaluated the success of the European Union's 12-year-long EUROTRA project to produce a multilingual MT system among the (then nine) official languages of the Union.[1] Attending

[1] Danzin, A., Allén, S., Coltof, H., Recoque, A., Steusloff, H. and O'Leary, M. (1990) 'Eurotra Programme Assess-

a session of the management committee which oversaw EUROTRA, I was perplexed to note that there seemed to be a radical misunderstanding between two sets of delegates: it was accepted that the project had not delivered the functioning system which had been the goal of the project; but was the report as a whole supportive or dismissive of EUROTRA's work? Did it suggest that more work should be undertaken, or the whole project abandoned as a failure? Broadly, the delegates split along language lines, the Romance-language speakers taking the report as more positive.

As it happened, the report had been written in French, but many of the committee had only read the English translation. On a crucial summary page, I discovered that the report had characterized the project's work as 'insuffisant', whereas the English version had translated this as 'inadequate'.

Arguably, no mistake had been made by the translator, in truth-conditional meaning, or even in style: when quality rather than quantity is being judged, it is much more natural in English to say 'inadequate' than 'insufficient'. But what a difference in connotation! What is called *insufficient* naturally needs to be supplemented, but what is termed *inadequate* is usually being roundly condemned. There could hardly be a clearer example of the treacherous nature of translation, even by the wise for the wise.

But what of MT itself? Have its results been insufficient or inadequate? It is very hard to give a final decision, although it is fairly clear that one concept which underlay most of the early work was basically inadequate.

The original rule-based models of MT which dominated research until the 1990s were essentially attempts to automate the "grammar-translation" approach to language learning. The syntactic rules of the various languages could be represented and programmed, and translation equivalents could be stipulated for lexical items, and for the semantic content of the various constructions. Proper names required access to vast encyclopaedias and gazetteers, seemingly never complete. The systems got larger and larger, and more cumbersome, harder to direct effectively.

Another response which became popular in the 1990s was to increase the role of machine intelligence, allowing inference engines to derive their own rules from exposure to vast amounts of translation equivalence data. This was computer equivalent of the "natural" method of language learning, essentially waiting for competence to arise unconsciously from massive exposure to language data. Perhaps the problems of performance here would ultimately yield as computers got exponentially faster and cheaper.

Yet systems remained lacking any general models which could represent the meaning of texts in the writer's or the reader's understanding, as they flitted from text to text or context to context. Nor was there any general means of selecting appropriate equivalents when language was used metaphorically. It seemed to prove that in practice, it was impossible to divorce the syntactic part of language processing from modelling the meaning of particular texts.

While this technical struggle continued unabated, the actual users of machine translation were devising their own *pis aller*, their own make-do approaches to handling what was available. The technology has begun to come into its own as a support system for human translators, allowing them to evade drudgery of repetitive translation and dictionary look-up. And the field of application has also been transformed by the vast quantities of foreign language text that are now available across the Internet. Automatic systems are proving useful aids to web-surfers, looking for relevant content in foreign disguise, rather than for clean translations of specific documents.

The fact that the technology is already being used serendipitously (rather than developed) by informal and linguistically-informed users is a first sign, I believe, of the actual future that awaits MT, and it is not an inglorious one.

The reason for the chronic dissatisfaction with MT's performance (especially among monolingual Anglophones, one may say) is that it has always been approached from a monolingual point of view, as a tool that is supposed to eliminate language barriers – i.e. as a means of converting all the alien codes into some readily understandable home language. This is the true in-

ment Report', Commission of the European Communities, DG-XIII, March 1990. French original as: Rapport Danzin : Document COM (90) 289 final.

adequacy in our traditional approach to MT. It is comparable to the lingua franca solution to multilingualism: let us find a common means – be it Latin, be it English, be it Esperanto – in which all the languages' texts' meanings can be represented. But a lingua franca is a practical solution in terms of a single language. MT has failed to do anything comparable, at least consistently, or reliably, or at a standard where the user familiar with English (or whatever target language is being attempted) is well satisfied.

But even in the forms currently available over the Internet, MT (and many other ad-hoc devices) already provide a vast number of tools to access and penetrate texts in unknown languages. It is debatable whether this is truly translation, and in many cases, the help is only accessible to those with a partial knowledge of the source language. But it does mean that, increasingly, partial understanding is becoming available, of texts that would in the past have been totally closed books.

Another personal anecdote may illuminate the situation that is emerging. According to Wikipedia, I "can functionally speak 26 different languages", this improbable claim having somehow emerged from the publicity department at Bloomsbury USA when they were designing the paperback jacket for my book *Empires of the Word*. This is harder to disprove than you might think, since paired with 'functionally' the verb 'speak' seems to be meant as equivalent to 'have command of'. I cannot know precisely which languages are intended here, but it is true that I have derived useful, and true, information from at least that many languages while working on that book and others. I cannot 'speak Chinese', but I was able to provide a phonetic transcription of texts from Confucius's *Analects*. Using other materials from the Internet I could gloss passages of Akkadian cuneiform and Egyptian hieroglyphs, locate relevant text in Sumerian, Persian and Portuguese, apply dialect changes to and parse Mexican Nahuatl and Palestinian Aramaic. In none of these languages can I boast any sort of fluency. But, in sum, my point is this: if you embrace the presence of foreign languages, and are interested enough to try to come to grips with them, more and more you will find the wherewithal to do so available to you (usually free of charge) on the Internet.

The set of language tools of which MT is a leading member are not available as a seamless suite which enables English users to look though the obscuring dark glass of foreign language to their crystal-pure meaning beneath, even if, here and there, web-page translation may in some cases be good enough to give this illusion. They are not, and cannot be, the realization of the monolingual dream of MT. But they are very much better than nothing, and – coupled with the right attitude to the point and value of foreign languages – they may be crucial aids to inter-lingual communication.

It is possible to look ahead into this dynamically improving, and enriching, world of inter-lingual electronic media. Just as the print-revolution – and various other social revolutions associated with urbanization – changed the ground-rules of communication among Europeans in the 16th century, so modern electronic technology is set to change the ancient need for a single lingua franca for all who wish to participate directly in the main international conversation. In brief, if electronics can remove the requirement for a human intermediary to interpret or translate, the frustrations of the language barrier may be overcome without any universal shared medium beyond compatible software. Recorded speeches and printed texts will become virtual media, accessible through whatever language the listener or speaker prefers. Machine translation, and language technologies more generally, may yet be very useful to minority languages, promoting and extending both their use and their status.

5 Will there be a Lingua franca?

We turn now to our third question: is the future course of globalization, as we currently perceive it, likely to call for a lingua franca in the long-term, whether it will be English, or some successor to it?

First of all, we can note that the forces making for the spread of English will soon peak, and the sequel will be a long retrenchment, as auxiliary English comes to be used less widely. Power, prestige, position, population, even practicality will never again favour English as they have in the 19th, 20th and early 21st centuries. If the world system remains dynamic, English will very much need to look to its laurels.

English does not even have all the advantages of position that Latin once had. Unlike Latin, once peerless in the world it knew, it does have com-

petitors – vernacular languages with hundreds of millions of speakers and intercontinental spread; and it has peaked in an age before some of their home populations have even reached their economic prime, China, India, Indonesia, Brazil, perhaps even Russia.

It will be strange if a country like India stays loyal to English once there is any serious trickle-down of its new and growing wealth. Already, its objective to double higher education by 2015 (to 15% of the age cohort) is putting pressure on the proportion educated in English. There is an issue here to be resolved, even if the outcome is not clear in advance. Perhaps – like Latin America in the 19th century – it will hold on to the language of its former colonists, and content itself paying lip-service to *indigenismo*, its heroic native roots. But regional languages are entrenched in the government of India, as they never were in Latin America: more likely, as in early modern Europe, it will be the elite language which has to yield. The bonds that tie India to English are far weaker than those of tradition and sentiment which once tied Europe to Latin.

It is often assumed that power politics and the global competition among great states will naturally be reflected linguistically. Hence the current international ubiquity of English is seen as a reflection of US 'unipolarity'. If this is doomed to pass, then it must, it is presumed, be followed by some other common language. The choice falls most obviously on Chinese, since this is already the world language with most speakers, and on current trends the Chinese economy is growing to be the largest in the world. Certainly the international importance of Chinese is very likely to grow, and as the Chinese become richer and more influential internationally, their concern to participate in the world on terms set by Anglo-Saxons will diminish. There is already evidence of this, highly predictable, change. The 2008 Pew Global Attitudes Survey in China reported that "Most Chinese (77%) agree that 'children need to learn English to succeed in the world today,' ... down substantially from 2002, when 92% agreed with this view."

However, this is only a small part of the coming changes. There are many parts of the world where English is not part of the national tradition, and they include the main countries about to increase in population size (sub-Saharan Africa, the Middle East) or relative wealth and influence (China, Russia, Brazil). Such a world is moving not to English or monolingualism, but it is hard to choose among these contenders for future linguistic influence. Very likely, the world is moving towards a much more multilingual, diverse, and potentially incalculable future.

6 Conclusion

But when technological ground is continually being ploughed up, there is cope for interesting new crops to germinate and flourish. Radical multilingualism may be one such crop, in a field-system (or a jungle) of pervasive digital technology. And monolingualism – privileging the stale over the fresh, and the few over the many – may well be an ideology whose time is passing.

Adaptable, Community-Controlled, Language Technologies for Language Maintenance

Lori Levin
Language Technologies Institute
Carnegie Mellon University
lsl@cs.cmu.edu

Abstract

Endangered languages may require more flexible language technologies than stable ones because they may not be standardized and they may be in a cycle of losing, replacing, and borrowing vocabulary and grammar. This paper argues that the coverage and content of language technologies should be in the hands of the speech community, and that it needs to be adaptable and learn from users. This calls for new approaches, possibly based on active learning to allow the language technologies to be as flexible and changeable as languages generally are. The paper also addresses ways in which the development of a machine translation system can be initiated when resources are scarce, including the experience of the AVENUE project with Mapudungun (Chile) and Iñupiaq (Alaska).

1 Introduction

All language professionals have been made aware of the plight of minor and endangered languages. In response, many language technologists have proposed methods for developing systems for languages that lack corpora and other resources. At the same time, speakers of endangered languages have become more aware of the potential of language technologies, bringing us to a point where we may ask ourselves how we can form partnerships that really help with language revitalization and design projects that are more than just exercises in research.

The AVENUE and LETRAS machine translation projects at Carnegie Mellon University[1] have had joint projects with two Native American language communities – the Mapuche in Chile and the Iñupiat in Alaska, speaking Mapudungun and Iñupiaq respectively. We have also had conversations with many others in order to find out what kind of language technologies can be useful.[2] As developers of machine translation, the AVENUE project team would like it to be the case that machine translation would magically revitalize a language by providing access to government, health care, education an the internet, all in the endangered language, thereby eliminating the need to use the surrounding language. However, we know that this position is naive or at least not viable in the near future. It is more likely that the goal should be a stable bilingual situation in which language technologies support the use of the endangered language without totally displacing the surrounding language.

Although promoting conversation with elders is probably the most desirable way to revitalize a language, it is also important for younger speakers to be able to communicate with each other using modern media. Margaret Noori[3] reports that her Ojibwe language students use text messaging, Facebook, Twitter, and adapted versions of video games in Ojibwe. In order to feel comfortable using these tools, non-native speakers need linguistic support. Welsh language revival is farther along in its support of modern media. The language technologies web page at Canolfan Bedwyr[4] lists lo-

[1]NSF grants IIS-0121631 and IIS-0534217.

[2]I would like to thank the following people for sharing their expertise and experience: Eliseo Cañulef, Rosendo Huisca, Edna MacLean, Lawrence Kaplan, Margaret Noori, Delyth Prys, and Per Langaard. I hope that I do not misrepresent their languages or communities. All mistakes are, of course, mine.

[3]http://www.umich.edu/~ojibwe/

[4]http://www.bangor.ac.uk/ar/cb/technolegau_iaith.php.en

Proceedings of the 13th Annual Conference of the EAMT, pages 8–11,
Barcelona, May 2009

calized operating systems and spelling checkers. Particularly important are tools to support texting on mobile phones[5] including access to dictionaries while texting. Speech technology is also important in language maintenance because it can allow speakers to say a word in order to learn how to spell it or spell a word to learn how to say it[6].

There is stable technology for many linguistic support tools such as spelling checkers, grammar checkers, on-line dictionaries, and speech recognition and synthesis. However, it may not be straightforward to adapt these tools to endangered languages. First, the languages may be typologically different from the ones that the technology was developed for. Polysynthetic languages are noticeably underrepresented in the world of language technologies. Second, the languages may not be standardized and there might be variation in everything from pronunciation to grammar. They may have to make up for lost words or make up new words for new things, or they may choose to borrow vocabulary from the surrounding language. They may also, unfortunately, be in the process of losing typologically rare features and gaining features of the surrounding language. Older speakers may have trouble accepting innovations in the language, but in the end, they realize that the language will only survive if it is allowed to change (Littlebear, 1999; Greymorning, 1999).

Setting aside the issue of documenting and preserving older, "correct" forms of a language, how can we as language technologists support a language that is in the process of rapid change and is being used by speakers who may not be completely fluent? There are many examples work heading in this direction. Three examples from Carnegie Mellon University are summarized here. Schultz and Black (Schultz et al., 2007) describe SPICE, a web based environment for building speech recognition and synthesis. It allows non-experts to enter initial data for training and confirm or disconfirm predicted pronunciations and spellings for additional data. Since the interface is easily usable by people who are not language technologists, the coverage and output of the system can be changed frequently. In statistical machine translation, Rogati (Rogati, 2009) uses active learning to reduce the amount of training data that is needed for domain adaptation by finding data that will have the most impact on performance. Font Llitjos (Font Llitjós, 2007), working with the AVENUE MT system, describes a Translation Correction Tool (TCT) that is operated by an MT user and allows the user to alter the behavior of an MT system. The user corrects erroneous translations and produces correct translations. The transfer rules that produced the erroneous translations are then automatically corrected.

[5]http://news.bbc.co.uk/1/hi/wales/4777933.stm

[6]http://www.abair.tcd.ie

2 The AVENUE Iñupiaq and Mapudungun Partnerships

The AVENUE machine translation framework developed at Carnegie Mellon University has been applied to many high resource and low resource languages, including two indigenous Western Hemisphere languages, Mapudungun (Chile) and Iñupiaq (Alaska). The full AVENUE framework includes several steps: (1) elicitation of data from native speakers, (2) automatic learning of transfer rules in a unification based synchronous grammar formalism based on the elicited data, (3)optional hand written transfer rules, (4) decoding, and (5) translation correction (as described above). For high resource languages, other techniques may be used such as statistical word alignment and extraction of syntactic phrases (Lavie, 2008; Hanneman and Lavie, 2009). These steps have not all been implemented for Mapudungun and Iñupiaq, but work is in progress.

2.1 Data Collection

Mapudungun and Iñupiaq are both low-resource languages in the sense that large corpora and dictionaries in electronic form are not available. (Although more resources are becoming available for Mapudungun.) Both languages have descriptive grammars, however, and there are native speakers who are linguists and language experts. Our partners include Edna MacLean and Larry Kaplan for Iñupiaq and Eliseo Cañulef and Rosendo Huisca for Mapudungun. The partner institutions were the Alaska Native Language Center (ANLC) at the University of Alaska at Fairbanks, the Universidad de la Frontera (UFRO) in Temuco, Chile, and the Chilean Ministry of Education.

We have proceeded with data collection in very different ways for Mapudungun and Iñupiaq based on resources that were available. Because we had a reasonable amount of funding for our initial work on Mapudungun, the UFRO team along

with Rodolfo Vega from Carnegie Mellon (CMU) collected and transcribed 170 hours of spoken Mapudungun (Monson et al., 2004). For Iñupiaq we have been pursuing other methods for acquiring data. The AVENUE elicitation corpus (Levin et al., 2006) consists of 3000 simple sentences illustrating grammatical features such as person, number, tense, aspect, animacy, and definiteness, as well as constructions such as relative clauses and questions. Edna MacLean translated the sentences into Iñupiaq and provided interlinear glosses. Some scanned texts were collected from ANLC and were typed by CMU undergraduates[7] resulting in a small corpus of 126K bytes. In addition, Shinjae Yoo at CMU is pursuing OCR with character n-grams for error correction as a method for increasing the size of the corpus.

2.2 Polysynthetic Morphology

Both Mapudungun and Iñupiaq are polysynthetic languages. Mapudungun stems can be simple or compounded. The compounds can involve noun incorporation, although this is becoming more rare, or verb compounding. After a verb stem there can be many closed class morphemes covering things like tense, aspect, agreement, passive and inverse voices, negation, and some adverbial and deictic meanings (Smeets, 1989; Zuñiga, 2000). Iñupiaq verbs also begin with stems followed by a large class of postbases, some of which have meanings related to English modal verbs and derivational morphemes (MacLean, 1993; MacLean, 1995). Inflectional morphemes follow the postbases. Iñupiaq has ergative case marking. Both Mapudungun and Iñupiaq have singular, dual, and plural number. Following are some examples of words in Mapudungun and Iñupiaq[8].

```
Mapudungun:
Treka  -l     -ke   -n.
walk   -CAUS  -HAB  -1.sg.IND
I usually make someone walk.

Mapudungun:
Kintu  -mara  -n.
hunt   -hare  -1sS/IND
I hunted (a/the) hare(s).


Inupiaq:
Imaqpaqaghaluaghniqsuq.

imaq  -qpak -qaq
water -big  -have
-kaluaq       -niq        -suq
-nevertheless -apparently -past.3.sg

Nevertheless it apparently had
a big body of water.
```

For both languages, building a morphological analyzer was a pre-requisite to doing any other work. The Mapudungun morphological analyzer was built by Carlos Fasola, Roberto Aranovich, and Christian Monson based on data provided by our partners at the Universidad de la Frontera (UFRO). The CMU team sorted the lexical items in the corpus by frequency, and the UFRO team provided morphological segmentation and glossing of the most common words. Because Mapudungun does not have much morpho-phonology at morpheme boundaries, the CMU team built a simple analyzer based on the legal order of morphemes (Aranovich, 2006). It does not take into account co-occurrence restrictions between morphemes and therefore produces spurious analyses of some morpheme sequences, which are then weeded out during machine translation. The Iñupiaq morphological analyzer is being implemented by Aric Bills based on published grammars by Edna MacLean (MacLean, 1993; MacLean, 1995) and is quite different from the Mapudungun system. Iñupiaq has extensive morphophonemic changes at morpheme boundaries. Inspired by Per Langaard's work on a morphological analyzer and spelling checker for Kalaallisut (Greenland), which is related to Iñupiaq, we decided to implement a transducer using the Xerox Finite State tools.

2.3 Machine Translation

We have not yet implemented automatic rule learning for Mapudungun or Iñupiaq. However, Roberto Aranovich (Aranovich, 2006; Font Llitjòs et al., 2005) has produced a hand-written transfer grammar for Mapudungun-to-Spanish MT. The system is currently small and is awaiting further development. It was tested on simple but unseen sentences from a textbook with about 65% accuracy after unkown vocabulary items were added. The main issue that was encountered was translation of negation, tense, and aspect morphemes. The AVENUE grammar formalism is synchronous, assuming corresponding source and target language rules applying in step with each other. Spaces were inserted between Mapudungun morphemes before translation so that each morpheme

[7] We would like to thank Ida Mayer, J. Eliot DeGolia, and Sai Venkateswaran for this work.

[8] The digraph *gh* is used in place of a dotted *g* in Iñupaiq.

would appear as a separate word, but we could not write transfer rules for every possible combination of Mapudungun morphemes and every possible corresponding sequence of Spanish words. Furthermore, in order to determine the tense of a Spanish sentence, it is sometimes necessary to look at multiple, non-adjacent morphemes in Mapudungun. The problem was solved using feature structures and unification, which are a part of the AVENUE transfer rule formalism. The Mapudungun morphemes were parsed and their features were stored in a feature structure until there was sufficient information to generate corresponding inflections, auxiliary verbs, negation, and adverbs in Spanish. In effect, transfer using synchronous grammars was not found to be useful for languages that are as different as Spanish and Mapudungun, but unification was found to be helpful.

3 Concluding Remarks

So far, this paper has recommended that language technologies for endangered languages be adaptable and community controlled in order to match the dynamic nature of language change and revitalization. Two additional issues related to endangered languages which are evident in our experience with Mapudungun and Iñupiaq are lack of electronic resources and typological divergences from major languages. It was suggested by Per Langaard that these problems could be solved by translating between related endangered languages. For example, Kalaallisut and Iñupaiq are related but are not equal in resources. Kalaallisut has newspapers, literature, and a textbooks for a full school curriculum. MT from Kalaallisut to Iñupiaq would probably produce more authentic and native sounding output than translation from English to Iñupiaq and could produce much needed literature and educational materials in Iñupiaq. Many other language families could also benefit from pooling resources in this way.

References

Aranovich, Roberto. 2006. *Handling of Translation Divergences in the Mapudungun/Spanish AVENUE Transfer Grammar and Lexicon*. Comprehensive Exam Paper, University of Pittsburgh.

Font Llitjòs, A., R. Aranovich, and L. Levin. 2005. Building machine translation systems for indigenous languages of latin america. In *Second Conference on the Indigenous Languages of Latin America (CILLA II)*.

Font Llitjós, Ariadna. 2007. *Automatic Improvement of Machine Translation Systems*. Ph.D. Thesis, Carnegie-Mellon University, School of Computer Science.

Greymorning, Stephen. 1999. Running the gauntlet of an indigenous language program. In *Revitalizing Indigenous Languages*, pages 6–16. Internet publication.

Hanneman, G. and A. Lavie. 2009. Decoding with syntactic and non-syntactic phrases in a syntax-based machine translation system. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*.

Lavie, Alon. 2008. Stat-xfer: A general search-based syntax-driven framework for machine translation. In Gelbuch, editor, *Computational Linguistics and Intelligent Text Processing*, pages 362–375. Springer, LNCS 4919.

Levin, L., J. Good, A. Alvarez, and R. Frederking. 2006. Parallel reverse treebanks for the discovery of morpho-syntactic markings. In *Proceedings of Treebanks and Linguistic Theories*.

Littlebear, Richard. 1999. Some rare and radical ideas for keeping indigenous languages alive. In Reyner, J., G. Cantoni, R.N. St. Clair, and E. Parsons Yazzie, editors, *Revitalizing Indigenous Languages*, pages 1–5. Internet publication.

MacLean, Edna. 1993. *North Slope Inupiaq Grammar: First Year (Revised)*. Alaska Native Language Center.

MacLean, Edna. 1995. *North Slope Inupiaq Grammar: Second Year (Revised)*. Alaska Native Language Center.

Monson, C., L. Levin, R. Vega, R. Brown, A. Font Llitjos, A. Lavie, C. Carbonell, E. Canulef, and R. Huisca. 2004. Data collection and analysis of mapudungun morphology for spelling correction. In *LREC*.

Rogati, Monica. 2009. *Domain Adaptation of Translation Models for Multilingual Applications*. Ph.D. Thesis (in progress), Carnegie-Mellon University, School of Computer Science.

Schultz, T., A. Black, S. Badaskar, M. Hornyak, and J. Kominek. 2007. Spice: Web-based tools for rapid language adaptation in speech processing systems. In *Interspeech*.

Smeets, I. 1989. *A Mapuche Grammar*. Ph.D. Thesis, University of Leiden.

Zuñiga, F. 2000. *Mapudungun*. Muenchen: Lincom Europa.

Character-based PSMT for Closely Related Languages

Jörg Tiedemann
Information Science
University of Groningen
PO Box 716
9700 AS Groningen, The Netherlands
j.tiedemann@rug.nl

Abstract

Translating unknown words between related languages using a character-based statistical machine translation model can be beneficial. In this paper, we describe a simple method to combine character-based models with standard word-based models to increase the coverage of a phrase-based SMT system. Using this approach, we can show a modest improvement when translating between Norwegian and Swedish. The potentials of applying character-based models to closely related languages is also illustrated by applying the character model on its own. The performance of such an approach is similar to the word-level baseline and closer to the reference in terms of string similarity.

1 Introduction

Closely related languages such as Norwegian and Swedish have many features in common. There are obvious similarities not only structurally but also lexically. Many differences are mainly due to writing conventions or consistent changes at the morpheme level. These facts should be beneficial for automatic translation especially for data-driven approaches. However, appropriate training material is often not available for such related languages, at least not in large amounts. This is due to the fact that speakers of these languages easily understand each other without switching to the foreign language and many documents are distributed in the original language only even in the neighboring countries. Still, there is a need for translation even for closely related language pairs as we can see in the Scandinavian situation. There are many types of textual data that have to be translated between languages such as Swedish, Norwegian and Danish ranging from movie subtitles, news to tourist information and others.

Due to the lack of training data for, e.g., Swedish-Norwegian a standard approach using phrase-based statistical machine translation faces the problem of handling unknown words probably more than, for example, the official EU languages for which sufficient amounts of training data is available. However, many of them (not only names) will actually be very similar to their translations. In this paper, we investigate the use of character-based PSMT models to translate such unknown words in order to improve the coverage of the MT system. In this way, we take Weaver's decoding idea to the extreme – translating foreign words as sequences of encoded characters. This approach has already been applied to another pair of closely related languages, Spanish and Catalan (Vilar et al., 2007). Our work mainly follows their approach. However, we use different settings and techniques for training our character-based model and also compare the various setups and their impact on translation quality.

The paper is organized as follows: First we will briefly mention related work. Thereafter, we describe the character-based model we will apply in the experiments discussed in the subsequent section. Finally we will summarize our study with some discussion and conclusions.

2 Related Work

As mentioned earlier, character-based SMT has already been applied to Spanish and Catalan (Vilar et al., 2007). Their letter-based system showed a quite acceptable performance and they concluded

Proceedings of the 13th Annual Conference of the EAMT, pages 12–19,
Barcelona, May 2009

that this technique is especially useful when training material is scarce. They also demonstrate a possible combination of letter-based and word-based models and obtained modest improvements in terms of BLEU scores.

Other solutions for the translation of special types of unknown words have been described in various articles. For example, the translation of named entities is discussed in (Chen et al., 1998; Al-onaizan and Knight, 2002). The treatment of compound words is discussed in (Koehn and Knight, 2003). Another idea for translating unknown words using analogical learning has been proposed by (Langlais and Patry, 2007). In their approach proportional analogies between strings are used to solve analogical equations to retrieve translations of previously unseen terms. The use of phrase-based statistical machine translation on the character level has already been described in (Matthews, 2007). In their work, these models are applied to the task of machine transliteration of Chinese-English and Arabic-English. Similar techniques can also be applied to languages using the same writing system in order to cover spelling differences of names even across related languages (Tiedemann and Nabende, submitted).

3 Character-based PSMT

Phrase-based statistical machine translation (PSMT) can be seen as one of the current state-of-the-art methods in data-driven machine translation. Due to the availability of tools such as Moses (Koehn et al., 2007) and GIZA++ (Och and Ney, 2003) this approach has received a lot of attention in the research community. Phrases in PSMT are usually defined as word N-grams and phrase translation models are estimated from word aligned parallel corpora. However, it is straightforward to apply the same tools used for training word-level PSMT models to train models on a different kind of segmentation level. For example, splitting sentences into character sequences makes it possible to train character based PSMT models in which phrases refer to character N-grams. The same applies for N-gram based language models which can be trained in a similar fashion on the character level. This is exactly the technique that has been applied in (Matthews, 2007) for transliteration and in (Vilar et al., 2007) for translation. In translation, the general assumption is that, similar to the transliteration task, many correspondences between lexical items of related languages can be explained on the character level. However, different to the transliteration approach we probably should not disable reordering as this can be important to capture consistent character movements. Furthermore, character sequences in the phrase table may often correspond to entire words and word phrases. Using reordering in the usual way we can still model phrase movements as in word based settings. In our experiments we will have a look at different settings for reordering in order to see the effect of these parameters.

The process of training character-based PSMT models includes the following steps: First the training data has to be split into character sequences. Important is to treat whitespace characters in a special way in order to keep the information of word boundaries in the data. We simply use the underscore character to replace whitespace characters. Consider the following example to illustrate the format of our training data:

Swedish: - _ D e t _ r ä c k e r _ !
Norwegian: - _ D e t _ e r _ n o k _ !

After pre-processing training data in this way we can use the same procedure as training a word-level model but now on the character level:

- creating a language model of character N-grams from the target language side of the corpus
- cleaning the training data (which includes the removal of sentences longer than 40 characters!)
- aligning characters with GIZA++ (using standard settings for all models involved up to IBM 4)
- symmetrizing character alignments, extracting N-gram translations and estimating their translation probabilities
- tuning the model with an independent development set (also in the same format using character sequences).

The maximum length of 40 tokens per sentence is typically applied for efficiency reasons when aligning with GIZA++. Tokens in our setting refers to characters and the restriction to 40 characters is very unfortunate. A large portion of the

data will be discarded in this way, which is, of course, a serious problem for statistical MT. This problem has already been pointed out by (Vilar et al., 2007). Therefore, they used a different technique for estimating their character-level translation model. They first aligned the corpus at the word level, extracted aligned phrases according to this alignment and then trained the character-based model on those phrase pairs. In this way, the whole corpus can be used assuming that the word alignment and phrase extraction is (mainly) correct.

Fortunately, our data consists of rather short sentences and sentence fragments and, therefore, the reduction of the training corpus due to preprocessing is not as severe as for other types of material. However, we still loose a lot of training data and, therefore, we also apply the two-step procedure as proposed by (Vilar et al., 2007) to compare our results with that approach. Interesting here is especially if the additional training data compensates for possible alignment errors in the phrase extraction.

Hvor er Lamborghinien ? Var är Lamborghinin ?
Jeg er lei av den . Jag är less på den .
Klar til å tape ? Redo att förlora ?
- Hvilke løp har du vunnet ? - Vad har du vunnit för lopp ?
- Ingen . - Inga .
Slår du meg i limousinen til mora di ? Slår du mig i din mammas limousine ?
Ja , jeg slår deg i mors limousin som jeg har trimmet på ymse måter . Ja , jag slår dig i mammas limousine som jag har trimmat på diverse vis .
Så du bø være temmelig redd . Så du bör vara mycket rädd .
Ikke like redd som du når mora di høer det . Inte lika rädd som du när din mamma hör det .

Figure 1: Examples from the Norwegian-Swedish training data.

4 Experiments

4.1 Data

The data for training, tuning and testing our approach is taken from the OpenSubtitle corpus, which is part of the OPUS collection (Tiedemann, 2008). The corpus contains a fair amount of Norwegian-Swedish aligned movie subtitles – still very little with respect to the requirements of statistical MT. Here are some statistics of the data used in our experiments:

training data : two different sets:

word model: 142,654 sentence pairs
1,015,844 Norwegian tokens
990,431 Swedish tokens
character model: ($<= 40$ char/sentence)
108,380 sentence pairs
601,100 Norwegian tokens
595,208 Swedish tokens

development set: 500 alignment units

evaluation set: 500 alignment units

Note, that the aligned sentences/sentence fragments are rather short which is common in movie subtitles. The average length for the training data of the character-based model is even less. The tuning and test sets are used in all experiments and do not have any length restriction. Figure 1 shows some example alignments from our training data.

As we can see in this little sample, there are a lot of similarities between lexical items in Swedish and Norwegian. However, there are also various syntactic differences even though both languages are structurally very close related with each other. Hence, a character-based SMT model will probably not be powerful enough on its own (not even with very long phrases that correspond to words and word n-grams) to take care of these phenomena without a decent reordering model on the word/phrase level. Therefore, we focus on the combination of both, a word-level and a character-level model.

In our experiments, the language models (for both, word LM and character LM) are simply trained on the target language side of our parallel data. We also add more out-of-domain data coming from the Europarl corpus (Koehn, 2005) to test a larger language model also on the character level.

4.2 Evaluation

For evaluation we apply the common automatic measures BLEU and NIST using the target side of the test set as our reference data (hence, we only have one reference per sentence). BLEU and NIST

are computed in the common way at the word-level for all three approaches: word-level SMT, character-level SMT and the combined models. In addition we also look at the string similarity between the translation and the reference sentence. We use the longest common subsequence ratio (LCSR) for this purpose, which is defined as the length of the longest common character sequence of two strings divided by the length of the longer of the two strings. We use this measure only to complement the MT evaluation measures without claiming that higher LCSR scores correlate with more acceptable translations. In future, we would like to investigate if it actually is possible to measure translation quality on the character level as well (using, for example, LCSR) as compared to word error rates, which is frequently used in MT evaluation as well. Here, the assumption would be that words which look more like target language words than others would make translations more acceptable. Especially for unknown words, which are usually just copied from source to target, it could well be the case that a character-based translation that comes close to the correct translation is more acceptable than an untranslated source language item. However, it could also be even more disturbing to see a lot of non-sense words instead of foreign words.

Finally, we also want to look at the significance of some of our result. For this we computed BLEU scores for individual sentences in our test set and compared paired BLEU scores using the nonparametric Wilcoxon matched-pairs signed-ranks test. One problem with BLEU is that it automatically becomes zero if for one of the N-gram sizes no match is found. The chance of seeing such a problem is of course quite high when running on single sentences especially for larger N-grams. Therefore, it is sometimes useful to test BLEU significance for different maximum N-gram sizes.

4.3 Baselines

For all our experiments we applied the Moses toolkit in connection with GIZA++ (Och and Ney, 2003) for word alignment and IRSTLM (Frederico et al., 2008) for language modeling.

4.3.1 Word-based PSMT

For the baseline of word-level PSMT we used a 5-gram language model and a maximum phrase length of 7 words. The alignment heuristics was set to `grow-diag-final-and` and all other parameters for training, phrase extraction and tuning were the default ones. We trained two word-level models using the two different training sets: the "big" training corpus and the "small" corpus that has been reduced in length for the character-level models. The parameters of both models were tuned using the same development set of 500 sentence pairs. Table 1 shows the BLEU and NIST scores for both corpora with different reordering models.

	BLEU	NIST	LCSR
$word_{big}$			
monotone	0.5167	7.3784	0.7675
distance ($<= 6$)	0.5293	7.4596	0.7725
lexicalised ($<= 6$)	0.5273	7.4688	0.7744
monotone (grow)	0.5169	7.4049	0.7668
distance (+EP-LM)	0.5298	7.4650	0.7728
$word_{small}$			
distance ($<= 6$)	0.4968	7.2863	0.7620
lexicalised ($<= 6$)	0.5012	7.2952	0.7595

Table 1: Baseline PSMT models with different types of reordering. The setting $grow$ refers to the alignment heuristics used in combining GIZA++ alignments. Otherwise the standard $grow-diag-final-and$ is applied. *EP-LM* refers to the additional data from the Europarl corpus used for language modeling. The lexicalised reordering model uses the option *msd-bidirectional-fe*.

As expected, the scores for all measures are lower for the smaller training set than for the bigger one[1]. However, the differences are not very big considering that the smaller set only includes about half of the tokens of the bigger set. We can also see that reordering is still important also for closely related languages. However, the improvements are rather modest compared to monotone decoding. The lexicalised reordering model did not add to the performance in this case (except of a very modest improvement on the small dataset). Furthermore, the additional data from Europarl used for language modeling does not increase the performance significantly.

4.3.2 Character-based PSMT

The second type of baseline refers to applying the character-based model to the test set in order to see its potentials when used on its own. We do not

[1]We did not use all reordering models for the smaller corpus. We simply wanted to compare the basic settings only when applied with different amounts of training data.

expect any improvements compared to the word-level baseline especially due to the limited reordering possibilities and the danger in producing nonsense words. We used two different settings for the character-based approach: The first one uses exactly the same settings as the word-based PSMT models. For the second, we increased the maximum length of extracted phrases, the distortion limit and the n-gram size of the language model to 10. The results of both approaches are shown in table 2.

	BLEU	NIST	LCSR
$char_{standard}$			
distance ($<= 6$)	0.4769	7.1455	**0.8030**
lexicalised ($<= 6$)	0.4898	7.1678	**0.8065**
$char_{long}$			
monotone	0.4894	7.1834	**0.8036**
distance ($<= 6$)	0.4917	7.2033	**0.8049**
lexicalised ($<= 6$)	0.5007	7.2775	**0.8094**
distance (+EP-LM)	0.4790	7.1151	**0.8029**
$two - steps$			
monotone	0.4738	7.1005	**0.7983**

Table 2: Character-based PSMT. "long" uses longer phrases (character N-grams – maximum of 10) and a 10-gram language model. In the $two - step$ model we used the phrases extracted from the word aligned corpus using the *grow* alignment heuristics. Otherwise the settings from the $char_{long}$ model apply.

The models are tuned with the same data set as the word-based models (but, of course, split into character sequences). As we can see in table 2, the models with longer phrases and a large N-gram model perform considerably better than the standard models when used with the same type of reordering[2]. They actually perform equally well as the word-based models trained on the same amount of data, which is a very encouraging result. We can also see that reordering still has a positive effect. Even on the character level, reordering still seems to be useful even if the improvements are very modest. Looking at the LCSR scores, we can also see that we get very close to the reference translation in terms of string similarity. Actually the translations are closer to the reference corpus than the ones from the word-based models. However, this does not necessarily have to mean that they are more acceptable as translations.

Finally, we also tested the significance of the BLEU score differences between some of the character-based models and the corresponding word-based models. According to the Wilcoxon matched-pairs test the differences between the character-based model with long phrases and distance-based reordering ($char_{long}$-distance) and the corresponding word-based model ($word_{big}$-distance) is not significant ($p > 0.05$ for BLEU; computed with both, a maximum of 3 and 4 for the size of N-grams to be checked). The same applies to the models using monotone decoding ($p > 0.1$ for max-3-gram-BLEU and max-4-gram-BLEU). For the models with lexicalized reordering, BLEU score differences are weakly significant ($p < 0.05$) for matches up to 4-grams but not for max-3-gram-BLEU scores. This seems to indicate that we indeed get very close to the performance of word-level models for all settings tested.

4.3.3 Character-based PSMT with prior word alignment

As the last baseline, the two-step procedure using word alignment and phrase extraction first to create the training data for the character level model is presented at the bottom of table 2. The advantage here, as mentioned earlier, is that we use the entire corpus for estimating our model instead of restricting ourselves to sentences with a maximum of 40 characters. We used the *grow* alignment heuristics for the word alignment in the first step in order to obtain reliable phrase pairs. Using other heuristics where unaligned tokens are added in the final steps add too much garbage to the training data which seriously harms our character-based model. Using extracted phrases as training material increases the size tremendously. We used the standard phrase extraction implemented in Moses and obtained over 4.5 million phrase pairs after word alignment. This, of course, includes a lot of overlapping phrases extracted from the aligned corpus. This might harm the model and further investigations are needed to check the influence of phrase extraction on this approach. Looking at the results in table 2 we can actually see that there is no improvement to be measured when using the large phrase pair corpus for estimating the character model, at least with monotone reordering as we have tested here. We doubt that other

[2]We did not apply monotone reordering on the smaller set mainly because of time issues. However, we expect the same tendency also for this type of model. We also omit the setting with a larger language model for the first setup as it already fails for the second one.

reordering models would change the results significantly. We will certainly try that in future experiments.

In the next two sections we will now look at two ways of combining character-based and word-based models. In both cases, character-based models are used for unknown words only – the ones we cannot find in the vocabulary files of our word-level model.

4.4 Merging Training Data

The first idea of combining word and character-based models is to merge training data and to train a new global model using both types of data. For this purpose, we simply attached the training data of the character-based model to the training data of the word-based model and trained as usual. Tuning is than also done on a combination of word-level tuning data and character-level tuning data. Certainly, this solution is a bit ad-hoc and especially the confusion between normal one-character words and character level parameters is very disturbing.

For testing, we like to focus on the translation of unknown words with the character-based model whereas other parts of the sentence will be taken care by the word-level model. This will cause another confusion in the model which is related to the distortion parameters learned from data which is either split on word or character level (but not both in the mixed test case). Results of this approach ("split unknown") using our test set are shown in table 3.

	BLEU	NIST	LCSR
standard	0.4979	7.2171	0.7598
split unknown	0.4758	6.9652	0.7602

Table 3: PSMT with merged training data (character-level & word-level). We used standard settings for model estimation, i.e. distance-based reordering and grow-diag-final-and for alignment. "standard" treats unknown words in the usual way by simply copying them to the target language output. In "split unknown" unknown words are split into character sequences before translating.

The scores are very disappointing. First of all, training on the combined data sets decreases already the performance of standard word-level PSMT. This was to be expected due to the ambiguity between character-level data and single-character words as discussed earlier. More disappointing is the combined approach when splitting unknown words into character sequences. The model does not seem to cope well with the input mixture.

4.5 Prior Translation of Unknown Words

The second approach is a cascaded one of translating unknown words first using a character-based model and then translating the rest using a word-level model with the already translated words escaped. Fortunately, Moses supports XML markup for such an escape mode in which translations of certain words are specified with special markup. We use the "exclusive" mode in which these translations will be fixed and copied to the target language output. Table 5 shows the result of applying the two models in such a sequential combination. We compare two settings: one with the "big" word-level model and one with the "small" word-level model. For both cases we use the tuned settings and the settings of the "long" character-based model. We only used distance based reordering without additional data for language modeling. Unfortunately, lexicalised reordering does not seem to work in Moses with additional XML markup in the input.

	BLEU	NIST	LCSR
$char_{long} + word_{small}$	0.5062	7.3513	0.7670
$char_{long} + word_{big}$	**0.5364**	**7.5116**	0.7769

Table 5: Two-step translation: First the character-based models for translating unknown words and then translating sentences with the word-based models (translated unknown words escaped).

Here, we can see a slight improvement in all scores compared to corresponding word-level baselines. The improvements are rather modest but considering that we actually translate only 175 unknown words ($word_{small}$) and 139 unknown words ($word_{big}$) within the 500 test sentences with the character-based model this result is still encouraging. This improvement is also significant according to the Wilcoxon matched-pairs test for both max-3-gram-BLEU scores and max-4-gram-BLEU scores ($p < 0.05$) when compared to the corresponding word-level baseline which is reassuring.

Reference	word-level baseline	$char_{long}$	$char_{long} + word_{big}$
- Välbevandrad .	- Velbevandret .	- Välbevandrats .	- Välbevandrat .
Häll i blekmedlet så här ...	Häll i blekemiddelet så här ...	Töm i blekamedelet sådan ...	Häll i blekamedelet så här ...
Du måste utforska möjligheterna och sen göra ditt val .	Du måste undersöka möjligheter och bestämma dig .	Du måste utforskar möjligheterna och bestämma dig .	Du måste undersöka möjligheter och bestämma dig .
Håller du med ?	Håller du ?	Är du med ?	Håller du ?
Strunta i idiot- tvillingarna .	Skit i idiottvillingene .	Skit i håret idiottvillingarna .	Skit i idiottvillingarna .
Jag måste ta över familjeföretaget .	Jag måste ta över familiefirmaet .	Jag måste ta över familjefirman .	Jag måste ta över familjefirman .
Ska jag inbilla mig att han är drömprinsen ?	Ska jag inbilla mig att han är drømmeprinsen ?	Ska jag inbilla mig att han är drömprinsen ?	Ska jag inbilla mig att han är drömprinsen ?
Han kör så det ryker , men är långt efter .	Han kjøer så det åker , men ligger långt bakom .	Han köar så det ryker , den ligger långt bakom .	Han köer så det åker , men ligger långt bakom .
Du är sån distraktion som jag ville undvika .	Du är ett sånt forstyrrende element som jag ville undvika .	Du är ett sånt förstörande elemente som jag ville undvika .	Du är ett sånt förstörande element som jag ville undvika .
Det är en naiv skolflicksdröm .	Det är en naiv skolejentedrøm .	Det är en naiv skolflickadröm .	Det är en naiv skola identifiedröm .

Table 4: Example translations from the Norwegian-Swedish test set. The first column shows the reference translation and the second column includes the baseline translation using a standard word-level PSMT model. The third column contains the translations of the character model on its own and the last column shows the combined model with unknown word translation as a pre-processing step.

5 Discussion & Conclusions

In this paper, we investigated the use of character-based PSMT models for the translation between closely related languages. The main goal of this approach is to combine such character-level transformations with standard word-level models in order to support the translation of unknown words. In our experiments with Norwegian and Swedish the potentials of such an approach could be seen even when applying such a character-based model on its own. Using this model for unknown words only in a pre-processing step resulted in a slight improvement according to automatic evaluation measures such as BLEU and NIST. Some example translations from the test set are shown in table 4.

Here, we can see some interesting examples. Some of the character-level translations of unknown words are very close to the reference translation, for example "Välbevandrat" (reference: "Välbevadrad") and "skolflickadröm" (reference: "skolflicksdröm"). Others are actually also acceptable even though they are not in the reference translation (for example "familjefirman" – reference: "familjeföretaget" and "förstörande element" – reference: "distraktion"). Other character-level translations are not acceptable, such as "köar" (to queue) instead of "kör" (to drive).

Furthermore, we can definitely see that character-based translation for related languages can be applied to various kinds of unknown words. This makes it very different from machine transliteration for which similar models have been applied before. The phrase-based character model can actually take care of word level translations as well as we can see in the compound "skolflickadröm" translated from the Norwegian "skolejentedrøm". Here, the Norwegian "jente" as part of the compound is translated into the Swedish "flicka" which is most certainly not a cognate word. There are many of such examples in the actual data where the character model takes care of word-level translations. In our data we can find examples such as "rolig - lugn", "akkurat - precis", "greit - okej", "trenger - behöver" and "begynne - att börja".

Furthermore, we have seen that a character-base model is able to generalize over certain regular transformations such as suffix correspondences, for example in translations such as "klippene - klipparna" and "sonettene - sonetterna" or "klarte - klarade" and "mente - menade". Other quite regular character transformations can also be detected, such as "e" to "ä" ("der - där", "kveld - kväll", "er - är", "rett - rätt", "foreldrene - föräldrar"), "kjø" to "kö" ("kjøttet - köttet") or "sjo" to "tio" ("vaksinasjoner - vaccinationer" or "ambisjoner - ambitioner"). All these examples are taken from the actual translations found in our data. Of course, character transformation also leads to many mistakes. However, often these translations come very

close to the correct expressions or at least to a hypothesis that looks very much like the target language. The same applies to the combined method. In most cases, including character level translations produces sentences that look more like the target language than the ones including unknown source language words, even if they contain certain mistakes that often look like typos. Some cases, however, are far from being correct and may disturb the readability more than leaving the original words in the target language output. Some user oriented study should be carried out to formally evaluate this impression. We would also like to investigate other ways of combining character level knowledge with word-level models. For example, we might be able to recognize character-level regularities that can directly be used in a word level model.

An interesting task for future work would be to see if similar techniques may be applied to improve unknown word translation for more distant language pairs as well. This has been tried already for the translation of names for which statistical transliteration modules could be used. This work can easily be extended to include historical cognates and more recent loan words. Furthermore, the character-based translation approach might also be successful for other language pairs with differences in compounding. As we have discussed earlier, compounds, which otherwise would be unknown to the system, can be covered using character-based translation tables.

The main difficulties of applying character-based models to distant language pairs are firstly the recognition of cases in which these models succesfully can be applied (named entity/loanword recognition) and, secondly, the collection of large amounts of appropriate training data, which should include cognates, transliterated names and loan words only. For the coverage of compounds it is even more difficult to find appropriate training data especially because compounding is usually very productive. Simple approaches to compound splitting might still be more effective.

References

Al-onaizan, Yaser and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In *In Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL02*, pages 400–408.

Chen, Hsin-hsi, Sheng-jie Huang, Yung-wei Ding, and Shih-chung Tsai. 1998. Proper name translation in cross-language information retrieval. In *In Proceedings of 17th COLING and 36th ACL*, pages 232–236.

Frederico, M., N. Bertoldi, and M. Cettelo, 2008. *IRSTLM Language Modeling Toolkit, Version 5.10.00*. FBK-irst, Trento, Italy.

Koehn, Philipp and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of EACL*, pages 187–193.

Koehn, Philipp, 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*, Prague, Czech Republic.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Langlais, Philippe and Alexandre Patry. 2007. Translating unknown words by analogical learning. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 877–886, Prague, Czech Republic, June. Association for Computational Linguistics.

Matthews, David. 2007. Machine transliteration of proper names. Master's thesis, School of Informatics, University of Edinburgh.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Tiedemann, Jörg and Peter Nabende. submitted. Translating transliterations. In *Annual International Conference on Computing and ICT Research (ICCIR 2009)*.

Tiedemann, Jörg. 2008. Synchronizing translated movie subtitles. In *Proceedings of the 6th International Conference on Language Resou rces and Evaluation (LREC'2008)*, Marrakesh, Marocco.

Vilar, David, Jan-Thorsten Peter, and Hermann Ney. 2007. Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 33–39, Prague, Czech Republic, June. Association for Computational Linguistics.

TS3: an Improved Version of the Bilingual Concordancer TransSearch

Stéphane Huet, Julien Bourdaillet and **Philippe Langlais**
DIRO - Université de Montréal
C.P. 6128, succursale Centre-ville
H3C 3J7, Montréal, Québec, Canada
{huetstep,bourdaij,felipe}@iro.umontreal.ca

Abstract

Computer Assisted Translation tools remain the preferred solution of human translators when publication quality is of concern. In this paper, we present our ongoing efforts conducted within TS3, a project which aims at improving the commercial bilingual concordancer TransSearch. The core technology of this Web-based service mainly relies on sentence-level alignment. In this study, we discuss and evaluate the embedding of statistical word-level alignment.

1 Introduction

Although the last decade witnessed an impressive amount of effort devoted to improving the current state of Machine Translation (MT), professional translators still prefer Computer Assisted Translation (CAT) tools, among which *translation memory* (TM) systems and *bilingual concordancers*. Both tools exploit a TM composed of a *bitext*: a set of pairs of units (typically sentences) that are in translation relation. Whereas a TM system is a translation device, a bilingual concordancer is conceptually simpler, since its main purpose is to retrieve from a bitext, the pairs of units that contain a *query* (typically a phrase) that a user manually submits. It is then left to the user to locate the relevant material in the retrieved target units. As simple as it may appear, a bilingual concordancer is nevertheless a very popular CAT tool. In (Macklovitch et al., 2008), the authors report that TransSearch,[1] the commercial web-based concordancer we focus on in this study, received an average of 177 000 queries a month over a one-year period (2006–2007).

Figure 1 provides a screenshot of a session with the current concordancer TransSearch. A user submitted the multi-word query in keeping with to which the system responded with a webpage showing the first 25 pairs of sentences in the TM that contain an occurrence of the query. As can be observed, nothing in the target material retrieved is emphasized, which forces the user to read the examples retrieved until an appropriate translation was found.

Figure 1: Screenshot of TransSearch. Two of the first 25 matches returned to the user for the query in keeping with.

The main objective of the TS3 project is to automatically identify (highlight) in the retrieved material the different translations of a user query. Identifying translations offers interesting prospects for user-efficient interactions. Although the definitive look-and-feel of the new prototype is not settled yet, Figure 2 shows an interface where the user can consult the most likely translations automatically

[1] www.tsrali.com

Proceedings of the 13th Annual Conference of the EAMT, pages 20–27,
Barcelona, May 2009

identified. Of course, she can still consult the pairs of sentences containing the query, but can as well click a given translation to see related matches.

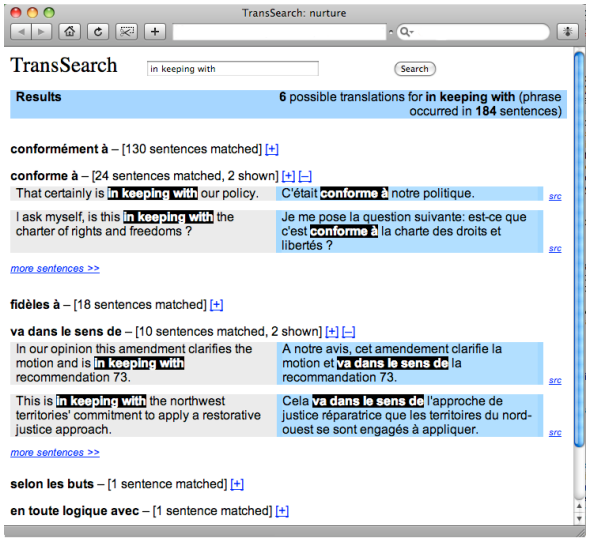


Figure 2: A hypothetical interface which exploits translation spotting.

The reminder of this paper is organized as follows. We first describe in Section 2 the translation spotting technique we implemented. Since translation spotting is a notoriously difficult problem, we discuss two novel issues that we think are essential to the success of a concordancer such as `TransSearch`: the identification of erroneous alignments (Section 3) and the grouping of translation variants (Section 4). We present the data we used in Section 5 and report on experiments in Section 6. We conclude our discussion and propose ongoing avenues in Section 7.

2 Transpotting

Translation spotting, or *transpotting*, is the task of identifying the word-tokens in a target-language (TL) translation that correspond to the word-tokens of a query in a source language (SL). It is therefore an essential part of the `TS3` project. We call *transpot* the target word-tokens automatically associated with a query in a given pair of units (sentences). For instance in Figure 2, `conformément à` and `va dans le sens de` are two of the six transpots displayed to the user for the query `in keeping with`.

2.1 Word Alignment

As mentioned in (Simard, 2003), translation spotting can be seen as a by-product of word-level alignment. Since the seminal work of (Brown et al., 1993), statistical word-based models are still the core technology of today's Statistical MT. This is therefore the alignment technique we consider in this study.

Formally, given an SL sentence $S = s_1...s_n$ and a TL sentence $T = t_1...t_m$ in translation relation, an IBM-style alignment $a = a_1...a_m$ connects each target token to a source one ($a_j \in \{1, ..., n\}$) or to the so-called NULL token which accounts for untranslated target tokens, and which is arbitrarily set to the source position 0 ($a_j = 0$). This defines a word-level alignment space between S and T whose size is in $O(m^{n+1})$.

Several word-alignment models are introduced and discussed in (Brown et al., 1993). They differ by the expression of the joint probability of a target sentence and its alignment, given the source sentence. For computational reasons, we focus here on the simplest form, which corresponds to IBM models 1&2:

$$p(t_1^m, a_1^m | s_1^n) = \prod_{j=1}^{m} \sum_{i \in [0,n]} p(t_j | s_i) \times p(i | j, m, n)$$

where the first term inside the summation is the so-called transfer distribution and the second one is the alignment distribution.

Given this decomposition of the joint probability, it is straightforward to compute the so-called Viterbi alignment, that is, the one maximizing the quantity $p(a_1^m | t_1^m, s_1^n)$. This approach often produces discontiguous alignments, which poses problems in practice. Furthermore, most of the queries in our logfile are contiguous ones, therefore we expect their translations to be contiguous as well.

2.2 Transpotting Algorithm

In order to enforce contiguous transpots, we implemented a variant of the transpotting algorithm initially proposed by Simard (2003), and which shares close similarities with phrase extraction technique described in (Venugopal et al., 2003). The idea is to compute for each pair $\langle j_1, j_2 \rangle \in [1, m] \times [1, m]$, two Viterbi alignments: one between the phrase $t_{j_1}^{j_2}$ and the query $s_{i_1}^{i_2}$, and one between the remaining material in the sentences $\bar{s}_{i_1}^{i_2} \equiv s_1^{i_1 - 1} s_{i_2+1}^n$ and $\bar{t}_{j_1}^{j_2} \equiv t_1^{j_1 - 1} t_{j_2+1}^m$. This method, which finds the translation of the query

according to:

$$\hat{t}^{\hat{j}_2}_{\hat{j}_1} = \underset{(j_1,j_2)}{\text{argmax}} \begin{cases} \max_{a^{j_2}_{j_1}} p(a^{j_2}_{j_1}|s^{i_2}_{i_1}, t^{j_2}_{j_1}) \\ \times \\ \max_{\bar{a}^{j_2}_{j_1}} p(\bar{a}^{j_2}_{j_1}|\bar{s}^{i_2}_{i_1}, \bar{t}^{j_2}_{j_1}) \end{cases},$$

has a complexity in $O(nm^3)$. It ranked first among several other alternatives we investigated (Bourdaillet et al., 2009).

2.3 The Need for Post-processing

Frequent queries in the TM receive numerous translations by the previously described transpotting process. Figure 3 illustrates the many transpots returned by our transpotting algorithm for the query `in keeping with`. As can be observed, some transpots (those marked by a star) are clearly wrong (*e.g.* `à`), many others (in italics) are only partially correct (*e.g.* `conformément`). Also, it appears that many transpots are indeed very similar (*e.g.* `conforme à` and `conformes à`).

`conforme à` (45)	`conformément à` (29)
`à`* (21)	`dans`* (20)
...	
`conforme au` (12)	`conformes à` (11)
`avec`* (9)	*`conformément`* (9)
...	
`correspond à` (1)	`respectent` (1)
`d'actualité`* (1)	`gestes en`* (1)

Figure 3: Subset of the 273 different transpots retrieved for the query `in keeping with`. Their frequency is shown in parentheses.

Since in `TS3` we want to offer the user a list of retrieved translations for a query, strategies must be devised for bypassing alignment errors and delivering as many as possible translation variants to the user. We investigated two avenues in this study: detecting erroneous transpots (Section 3) and merging together variants of the same canonical translation (Section 4).

3 Refining Transpotting

We investigated the learning of classifiers trained to distinguish good transpots from bad ones. We tried several popular classifiers:[2] a *voted-perceptron* algorithm (Freund and Schapire, 1999) which has been reported to work well in a number of NLP tasks (Collins, 2002); a *support vector machine* (SVM), commonly used in supervised learning (Cristianini and Shawe-Taylor, 2000); a *decision stump*, a very simple one-level decision tree; *AdaBoost* using a decision stump as weak classifier (Freund and Schapire, 1996); and a *majority voting* classifier between a voted-perceptron, an SVM and AdaBoost (Kittler et al., 1998).

Each classifier was trained in a supervised way thanks to an annotated corpus we devised (see Section 5). We computed three groups of features for each example, that is, each query/transpot pair (q, t). The first group is made up of features related to the size (counted in words) of q and t, with the intuition that they should be related. The second group gathers various alignment scores computed with word-alignment models (min and max likelihood values, etc.). The last group gathers clues that are more linguistically flavored, among which the ratio of grammatical words in q and t, or the number of prepositions and articles. In total, each example is represented by at most 40 numerical features.

4 Merging Variants

Once erroneous transpots have been filtered out, there usually remain many translations for a given query. For instance, the best classifier we trained identified 91 bad transpots among the 273 candidat ones. Among the remaining transpots, some of them are very similar and are therefore redundant for the user (see Figure 3). This phenomenon is particularly acute for the French language with its numerous conjugated forms for verbs. Another problem that often shows up is that many transpots differ only by punctuation marks or by a few grammatical words.

Of the 182 transpots surviving the filter, we estimate that no less than 37 interesting *canonical* translations exist for the query `in keeping with`. Therefore, it is important from the user perspective to identify them. We investigated ways to merge together close variants. This raises several difficulties. First, the transpots must be compared together, which represents both a tricky and time consuming process. Second, we need to identify groups of similar variants. We describe our solutions to these problems in the sequel.

[2]We used Weka in our experiments `www.cs.waikato.ac.nz/ml/weka`.

4.1 Word-Based Edit Distance

A word-level specific edit distance was empirically developed to meet the constraints of our application. Different substitution, deletion and insertion costs are set according to the grammatical classes or possible inflections of the words; it is therefore language dependent. We used an in-house lexicon that lists, for both French and English, the lemmas of each inflected form and its possible parts-of-speech.

A minimal substitution cost was empirically given between two inflected forms of the same lemma. A score has been engineered which increasingly penalizes in that order edit operations involving punctuation marks, articles, grammatical words (prepositions, conjunctions and pronouns), auxiliary verbs and lexical words (verbs, nouns, adjectives and adverbs).

4.2 Neighbor-Joining Algorithm

Comparing the transpots pairwise with the distance we defined is an instance of multiple sequence alignment, a well studied problem in bioinformatics (Chenna et al., 2003). We adopted the approach of progressive alignment construction. This method first computes the word-based edit-distance between every pair of transpots and stores the results in an edit-matrix. Second, a greedy bottom-up clustering method called *neighbor-joining* (Saiou and Nei, 1987) is conducted; it builds a tree by joining together either two transpots, that is two leaves of the tree, or a transpot and a node in the tree already aggregating several translations. At each step, the most similar pair is merged and added to the tree, until no transpot remains unaligned.

Finally, the neighbor-joining algorithm returns a tree whose leaves are transpots. Closest leaves in this tree correspond to the most similar variants. Therefore, clusters of variants can be formed by traversing the tree in a post-order manner. The transpots associated with two neighboring leaves and which differ only by grammatical words or by inflectional variants are considered as sufficiently similar to be merged into a single cluster. This process is repeated until all the leaves have been compared with their nearest neighbor and no more similar variants remain.

Figure 4 illustrates this process. The two neighbor transpots `conforme à` and `conformes à` are first grouped together, so are `conforme au` and `conforme aux`. Then, those two groups are merged into a single cluster. The transpot `correspondant à` being too different is not aggregated into this cluster.

Figure 4: Merging of close transpots.

4.3 Naive Joining Algorithm

We also implemented a conceptually simpler merging algorithm which relies on the frequencies of transpots. The algorithm compares the most frequent variant with all the other ones. Those that are close enough (according to our distance) are aggregated into a cluster. This process is iteratively applied on the remaining variants until no more cluster can be formed.

5 Corpora

5.1 Translation Memory

The largest collections in `TransSearch` come from the Canadian Hansards, that is, parallel texts in English and French drawn from official records of the proceedings of the Canadian Parliament. For our experiments, we indexed with Lucene[3] a TM comprising 3.3 million pairs of French-English sentences aligned at the sentence-level by an in-house aligner. This was the maximum amount of material we could train a statistical word-alignment model on, running the `giza++` (Och and Ney, 2003) toolkit on a computer equipped with 16 gigabytes of memory.

5.2 Automatic Reference Corpus

We developed a reference corpus (REF) by intersecting our TM with a bilingual lexicon, and some user queries. We used an in-house bilingual-phrase lexicon we collected over various projects, which includes 59 057 English phrases with an average of 1.4 French translations each. We extracted from the logs of `TransSearch` the 5 000 most frequent queries submitted by users to the system. 4 526 of those queries actually occurred in our TM, and of these, 2 130 had a sanctioned translation in our bilingual lexicon. We collected up to 5 000 pairs of sentences for each of those 2 130 queries,

[3] `http://lucene.apache.org`

leading to a set of 1 102 357 pairs of sentences, with an average of 517 pairs of sentences per query. For each of the 2 130 queries, the bilingual lexicon enabled us to extract a mean of 3.5 different transpots (and a maximum of 37). This results in a set of 7 472 different pairs of query/translation.

5.3 Human Reference

In order to train the classifiers described in Section 3, four human annotators were asked to identify bad transpots among those proposed by our transpotting algorithm. We decided to annotate the query/transpot pairs without their contexts of occurrence, which allows a relatively fast annotation process,[4] but leaves some cases difficult to annotate. For instance, in our running example, a transpot such as `conforme à` is straightforward to annotate, but others such as `dans le sens de` or `tenir compte de` gave annotators a harder time since both can be valid translations in some contexts. We ended up with a set of 531 queries that have an average of 22.9 transpots each, for a total of 12 144 annotated examples. We computed the inter-annotator agreement and observed a 0.76 kappa score, which indicates a high degree of agreement.

6 Experiments

6.1 Transpotting

For each of the 1 102 357 pairs of sentences of REF, we evaluated the ability of the transpotting algorithm described in Section 2.2 to find the reference translation $\hat{t}$ for the query q, according to recall and precision ratios computed as follows:

$$\text{recall} = |t \cap \hat{t}|/|\hat{t}| \quad \text{precision} = |t \cap \hat{t}|/|t|$$
$$\text{F-measure} = 2 \times |t \cap \hat{t}|/(|t| + |\hat{t}|)$$

where t is the transpot identified by the algorithm, and the intersection operation is to be understood as the portion of words shared by t and $\hat{t}$. A point of detail is in order here: since several pairs of sentences often contain the same given query/reference translation pair $(q, \hat{t})$, we first average for a given pair the ratios measured for all the occurrences of that pair in the reference corpus. Then, we average the scores over the set of all different pairs $(q, \hat{t})$ in the corpus. This avoids biasing our evaluation metrics toward frequent pairs in the REF corpus.[5]

[4] On the order of 40 seconds per query.

[5] Without this normalization, results would be increased by a range of 0.2 to 0.4 points.

	prec.	rec.	F-meas.
transpotting	0.30	0.60	0.38
transpotting + voting	0.37	0.76	0.46

Table 1: Transpotting results before and after filtering (REF). See next section for an explanation of line 2.

Our transpotting algorithm (see line 1 of Table 1) achieves a precision of 0.30, and a recall of 0.60. At a first glance, these figures might seem rather low. However, recall that our normalization prevents frequent queries that are often correctly aligned from being counted several times. Thus, this reinforces the score measured for infrequent queries, which in turn tend to be worse aligned. Also, we observed that very often the reference translation is a subset of the transpot found, which lowers precision. This is the case of the example shown in Figure 5.

`Une telle restriction ne s' inscrit pas` **`dans le sens des`** `pratiques actuelles.`

Figure 5: Transpot (underlined) and reference translation (in bold) for the query `in keeping with`.

6.2 Training Classifiers

As described in Section 3, we trained various classifiers to identify spurious transpots, representing an example (a query/transpot pair) by three kinds of feature sets. All these variants plus a few challenging baselines are evaluated according to the ratio of Correctly Classified Instances (CCI). Since in our application we are interested in filtering out bad transpots, precision, recall and F-measure rates related to this class are computed as well.

We report in Table 2 the figures we measured by a 10-fold stratified cross-validation procedure. To begin with, the simplest baseline we built (line 1) classifies all instances as good. This results in a useless filter with a CCI ratio of 0.62. A more sensible baseline —that we engineered after we investigated the usefulness of different feature sets— classifies as bad the transpots whose ratio of grammatical words is above 0.75. It is associated with a CCI ratio of 0.78 (line 2).

We started by investigating the voted-perceptron

			Bad		
Classifier	Features	CCI	precision	recall	F-measure
Baseline: all good		0.62	0.00	0.00	0.00
Baseline: grammatical ratio > 0.75		**0.78**	0.88	0.49	**0.63**
Voted-Perceptron (VP)	size	0.73	0.75	0.47	0.58
	IBM	0.78	0.69	0.78	0.73
	grammatical	0.79	0.88	0.52	0.65
	all	0.83	0.81	0.73	0.77
SVM	all	0.83	0.84	0.70	0.76
Decision Stump		0.81	0.77	0.70	0.74
AdaBoost		0.83	0.71	0.83	0.76
Majority-Voting (VP+SVM+AdaBoost)		**0.84**	0.84	0.71	**0.77**

Table 2: Performance of different algorithms for identifying bad transpots.

and the contribution of each feature sets on its performance.[6] When the voted-perceptron is trained using only one set of features, the one making use of the grammatical features obtains the best CCI ratio of 0.79 and an F-measure of 0.65. Even if the configuration based on IBM model 2 word alignment scores obtains a slightly inferior CCI ratio of 0.78, it has a much higher F-measure of 0.73 and can be considered as the best feature set. When using all feature sets, the voted-perceptron clearly surpasses the baseline with a CCI of 0.83 and an F-measure of 0.77. It should be noticed that while the best baseline has a better precision than the best voted-perceptron, precision and recall are more balanced for the latter. Because it is not clear whether precision or recall should be favored for the task of bad transpot filtering, optimizing the F-measure is preferred.

When training the other classifiers using all feature sets, no significant gain can be observed. Nevertheless the majority-voting classifier obtains the best CCI ratio of 0.84 and an F-measure of 0.77. The figures obtained by the decision stump, a one-level decision tree, are surprisingly high. The rule used by this classifier considers the minimal word alignment probability inside a Viterbi alignment based on an IBM model 2. At the very least, this confirms the interest of this feature set.

Once the best classifier had been obtained, *i.e.* majority-voting, we evaluated the impact of transpotting filtering against the REF corpus. Results are shown in Table 1 (line 2). We observe a significant gain in F-measure which increases from 0.38 to 0.46. The higher gain is in recall, from 0.60 to 0.76. Referring to the example of Section 6.1, this means that filtering eliminates too short transpots. Inspections revealed that short bad transpots, such as grammatical words, are frequently identified as bad by the classifier. This demonstrates the interest of filtering bad transpots.

6.3 Merging Variants

The interest of grouping together similar variants is clear from a user perspective. However, the granularity with which we should aggregate variants is not obvious.[7] We studied two approaches. The first method aims at grouping together transpots that differ by punctuation marks or that are inflectional variants of the same lemma. It is based on an edit distance, called D_1, which uses the same edit-costs for grammatical and lexical words. The second method groups together variants with looser constraints. It resorts to an edit distance, named D_2, that associates lower edit-costs with grammatical words than with lexical words.

From the transpots obtained for the 5 000 queries of the REF corpus (and filtered by our best classifier), this method leads to an average of 69 clusters per query (Table 3, columns 2 and 4), whereas there are on average 85 unique transpots per query (Table 3, column 1). The same level of grouping is observed for the two joining algorithms described in Section 4.3.

As expected, the use of D_2 dramatically reduces the number of clusters to an average of 45 per query (Table 3, columns 3 and 5). Contrary to D_1, D_2 allows the merging process to gather similar variants such as `sur des années` and

[6] Similar results concerning the different feature set have been observed for the other classifiers and are not presented here.

[7] This would certainly require tests with real users.

baseline	naive joining		neighbor-joining	
	D_1	D_2	D_1	D_2
85	69	45	69	45

Table 3: Average number of responses per query.

durant des années. However, it occasionally leads to erroneous groupings such as tout à fait (fully) and fait tout (do everything).

In what follows, we measure quantitatively the improvement from the point of view of the quality of the first responses suggested for each query.

Experimental Setup Table 4 shows the 5 most frequent transpots computed for two queries by the original transpotting algorithm (baseline) and obtained after grouping together variants (with the naive joining method). We observe the tendency of the baseline to propose inflectional variants of the same translation, while merging variants leads to more diversity, which is preferable since the number of variants that can be displayed in TransSearch without scrolling is limited. Indeed, we think that presenting the user with around 5 transpots and some sentences where they occurred is a good compromise (see Figure 2).

In order to simulate this, we measure in what follows the diversity of the best 5 transpots proposed by different methods. The baseline keeps the 5 most frequent transpots as returned by our transpotting algorithm, while the other methods allow for clustering the transpots. The 5 most frequent clusters are considered,[8] and the most frequent variant in each cluster is retained. Therefore each method delivers at most 5 transpots.

The best 5 transpots are considered as bags of unigrams, bigrams or trigrams and compared to reference translations turned also in bags of n-grams. All the words are lemmatized, and short words (less than 4 characters) are discarded as a proxy to remove grammatical words. For instance, the transpots returned by the baseline method in Table 4 for the first query are turned into {décrire, comme}.

Results The comparison of the generated bags-of-words with the reference ones is done by computing precision and recall. The reference used here is the resource described in Section 5.3. Table 5 reports results obtained with the metrics based on bags of n-grams without joining variants (line 1) and when using either the neighbor-joining algorithm (lines 2 and 3) or the naive method (lines 4 and 5). Their comparison shows an improvement in terms of F-measure for unigrams, bigrams and trigrams when variants are merged. If the precision slightly decreases for unigrams w.r.t. the baseline, a significant improvement is obtained especially with the edit distance D_2. These results are correlated with the more diversified translations obtained when variants are grouped together.

[8]The frequency of a cluster is the cumulative frequency of all the variants it groups.

7 Discussion

In this study, we have investigated the use of statistical word-alignment for improving the commercial concordancer TransSearch. A transpotting algorithm has been proposed and evaluated. We discussed two novel issues that are essential to the success of our new prototype: detecting erroneous transpots, and grouping together similar variants. We proposed our solutions to these two problems and evaluated their efficiency. In particular, we demonstrated that it is possible to detect erroneous transpots better than a fair baseline, and that merging variants leads to transpots of better diversity.

For the time being, it is difficult to compare our results to others in the community. This is principally due to the uniqueness of the TransSearch system, which archives a huge TM. To give a point of comparison, in (Callisson-Burch et al., 2005) the authors report alignment results they obtained for 120 selected queries and a TM of 50 000 pairs of sentences. This is several orders of magnitude smaller than the experiments we conducted in this study.

There are several issues we are currently investigating. First, we only considered simple word-alignment models in this study. Higher-level IBM models can potentially improve the quality of the word alignments produced. At the very least, HMM models (Vogel et al., 1996), for which Viterbi alignments can be computed efficiently, should be considered. The alignment method used in current phrase-based SMT is another alternative we are considering.

Acknowledgements

This research is being funded by an NSERC grant in collaboration with Terminotix.[9]

[9]www.terminotix.com

baseline	décrits	décrite	décrit	tel que décrit	comme l'a
naive joining D_2	décrits	prévu	comme l'a	tel que prescrit	comme le propose
baseline	s'est révélé	s'est avéré	s'est avérée	s'est révélée	a été
naive joining D_2	s'est révélé	s'est avéré	a été	s'est montré	a prouvé

Table 4: 5 most frequent responses for the queries `as described` and `has proven to be` when a joining method is used or not.

		unigrams			bigrams			trigrams		
		prec.	rec.	FM	prec.	rec.	FM	prec.	rec.	FM
baseline		0.93	0.45	0.61	0.82	0.35	0.49	0.68	0.30	0.41
naive joining	D_1	0.93	0.51	0.65	0.86	0.40	**0.55**	0.72	0.33	0.45
	D_2	0.90	0.57	**0.69**	0.79	0.40	0.53	0.72	0.33	0.45
neighbor-joining	D_1	0.93	0.50	0.65	0.86	0.41	**0.55**	0.72	0.34	**0.46**
	D_2	0.90	0.56	**0.69**	0.80	0.40	0.53	0.71	0.34	**0.46**

Table 5: Evaluation of quality of the variants merging process for the 5 most frequent groups retrieved for 531 queries.

References

Bourdaillet, J., S. Huet, F. Gotti, G. Lapalme, and P. Langlais. 2009. Enhancing the bilingual concordancer TransSearch with word-level alignment. In *22nd Conference of the Canadian Society for Computational Studies of Intelligence*, Kelowna, Canada.

Brown, P., V. Della Pietra, S. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

Callisson-Burch, C., C. Bannard, and J. Schroeder. 2005. A compact data structure for searchable translation memories. In *10th European Conference of the Association for Machine Translation (EAMT)*, pages 59–65, Budapest, Hungary.

Chenna, R., H. Sugawara, T. Koike, R. Lopez, T. J. Gibson, D. G. Higgins, and J. D. Thompson. 2003. Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Research*, 31(13):3497–3500.

Collins, M. 2002. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8, Philadelphia, PA, USA.

Cristianini, N. and J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press.

Freund, Y. and R. Schapire. 1996. Experiments with a new boosting algorithm. In *13th International Conference on Machine Learning (ICML)*, pages 148–156, Bari, Italy.

Freund, Y. and R. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

Kittler, J., M. Hatef, R. P.W. Duin, and J. Matas. 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239.

Macklovitch, E., G. Lapalme, and F. Gotti. 2008. TransSearch: What are translators looking for? In *18th Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 412–419, Waikiki, Hawai'i, USA.

Och, F. J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Saiou, N. and M. Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425.

Simard, M. 2003. Translation spotting for translation memories. In *HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and beyond*, pages 65–72, Edmonton, Canada.

Venugopal, A., S. Vogel, and A. Waibel. 2003. Effective phrase translation extraction from alignment models. In *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 319–326, Sapporo, Japan.

Vogel, S., H. Ney, and Tillmann C. 1996. HMM-based word alignment in statistical translation. In *16th Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark.

Estimating the Sentence-Level Quality of Machine Translation Systems

Lucia Specia*, Nicola Cancedda and Marc Dymetman
Xerox Research Centre Europe
Meylan, 38240, France
lucia.specia@xrce.xerox.com
nicola.cancedda@xerox.com
marc.dymetman@xerox.com

Marco Turchi* and Nello Cristianini
Department of Engineering Mathematics
University of Bristol
Bristol, BS8 1TR, UK
Marco.Turchi@bristol.ac.uk
nello@support-vector.net

Abstract

We investigate the problem of predicting the quality of sentences produced by machine translation systems when reference translations are not available. The problem is addressed as a regression task and a method that takes into account the contribution of different features is proposed. We experiment with this method for translations produced by various MT systems and different language pairs, annotated with quality scores both automatically and manually. Results show that our method allows obtaining good estimates and that identifying a reduced set of relevant features plays an important role. The experiments also highlight a number of outstanding features that were consistently selected as the most relevant and could be used in different ways to improve MT performance or to enhance MT evaluation.

1 Introduction

The notion of "quality" in Machine Translation (MT) can have different interpretations depending on the intended use of the translations (e.g., fluency and adequacy, post-editing time, etc.). Nonetheless, the assessment of the quality of a translation is in general done by the user, who needs to read the translation and the source text to be able to judge whether it is a good translation or not. This is a very time consuming task and may not even be possible, if the user does not have knowledge about the source language. Therefore, automatically assessing the quality of translations produced by MT systems is a crucial problem, either to filter out the low quality ones, e.g. to avoid professional translators spending time reading / post-editing bad translations, or to present them in such a way as to make end-users aware of the quality. This task, referred to as Confidence Estimation (CE), is concerned about predicting the quality of a system's output for a given input, without any information about the expected output.

CE for MT has been viewed as a binary classification problem (Blatz et al., 2003) to distinguish between "good" and "bad" translations. However, it may be difficult to find a clear boundary between "good" and "bad" translations and this information may not be useful in certain applications (e.g, the time necessary to post-edit translations).

We distinguish the task of CE from that of MT evaluation by the need, in the latter, of reference translations. The general goal of MT evaluation is to compare a machine translation to reference translation(s) and provide a quality score which reflects how close the two translations are. In CE, the task consists in estimating the quality of the translation given only information about the input and output texts and the translation process.

In this paper we consider CE for MT as a wider problem, in which a continuous quality score is estimated for each sentence. This could be seen as a proxy for MT evaluation, but without any form or reference information. This problem is addressed as a regression task, where we train algorithms to predict different types of sentence-level scores. The contribution of a large number of features is epxloited by using a feature selection strategy. We also distinguish between features that depend on the translation process of a given MT system and those that can be extracted given only the input sentences and corresponding output translations,

*L. Specia and M. Turchi contributed equally to this work.

Proceedings of the 13th Annual Conference of the EAMT, pages 28–35,
Barcelona, May 2009

and are therefore independent on MT systems.

In the remaining of this paper we first discuss the previous work on CE for MT (Section 2), to then describe our experimental setting (Section 3) and method (Section 4) and present and discuss the results obtained (Sections 5 and 6).

2 Related work

Early work on CE for MT aimed at estimating the quality at the word level (Gandrabur and Foster, 2003; Ueffing and Ney, 2005; Kadri and Nie, 2006). Sentence-level CE appears to be a more natural set-up for practical applications of MT. One should consider as real-world scenario for CE an MT system in use, which would provide to the user, together with each sentence translation, an estimate of its quality. If this estimate is in the form a numerical score, it sould also be viewed as a proxy to some automatic or manual metric, like NIST (Doddington, 2002) or 1-5 adequacy. Other estimates include the time that would be necessary to post-edit such translation, or simply a "good" / "bad" indicator.

Differently from MT evaluation, in CE reference translations are not available to compute the quality estimates. Therefore, CE approaches cannot be directly compared to the several recently proposed metrics for sentence-level MT evaluation that also use machine learning algorithms and sometimes similar features to those used in CE. For example, (Kulesza and Shieber, 2004) use Support Vector Machines (SVM) with n-gram precision and other reference-based features to predict if a sentence is produced by a human translator (presumably good) or by a MT system (presumably bad) (*human-likeness classification*). (Albrecht and Hwa, 2007a) rely on regression-based algorithms and features, like string and syntax matching of the translation over the corresponding references, to measure the quality of sentences as a continuous score. In (Albrecht and Hwa, 2007b), *pseudo-references* (produced by other MT systems) are used instead of human references, but this scenario with multiple MT systems is different from that of CE.

The most comprehensive study on CE at the sentence level to date is that of (Blatz et al., 2004). Multi-layer perceptrons and Naive Bayes are trained on 91 features extracted for translations tagged according to NIST and word error rate. Scores are thresholded to label the 5th or 30th percentile of the examples as "correct" and the remainder as "incorrect". Regression is also performed, but the estimated scores are mapped into the same classes to make results binary. The contribution of features is investigated by producing classifiers for each feature individually and for combinations of all features except one at a time. In both cases, none of the features is found to be significantly more relevant than the others. This seems to point out that many of the features are redundant, but this aspect is not investigated.

(Quirk, 2004) uses linear regression with features similar to those used in (Blatz et al., 2004) to estimate sentence translation quality considering also a small set of translations manually labeled as correct / incorrect. Models trained on this small dataset (350 sentences) outperform those trained on a larger set of automatically labeled data. Given the small amount of manually annotated data and the fact that translations come from a single MT system and language-pair, it is not clear how results can be generalized. The contribution of different features is not investigated.

(Gamon et al., 2005) train an SVM classifier using a number of linguistic features (grammar productions, semantic relationships, etc.) extracted from machine and human translations to distinguish between human and machine translations (*human-likeness classification*). The predictions of SVM, when combined to a 4-gram language model score, only slightly increase the correlation with human judgements and such correlation is still lower than that achieved by BLEU (Papineni et al., 2002). Moreover, as shown in (Albrecht and Hwa, 2007a), high human-likeness does not necessarily imply good MT quality. Besides estimating the quality of *machine* translations directly, we use a larger set of features, which are meant to cover many more aspects of the translations. These features are all resource-independent, allowing to generalize this method across translations produced by several MT systems and for different language-pairs.

Although our goal is very similar to that of (Blatz et al., 2004; Quirk, 2004), it is not possible to compare our results to these previous works, since we estimate continuous scores, instead of binary ones. We consider the following aspects as main improvements wrt such previous works: (a) evidence that is is possible to accurately estimate continuous scores, besides binary indicators,

which can be more appropriate for certain applications (e.g. post-edition time); (b) the use of learning techniques that are appropriate for the type of features used in CE (Partial Least Squares, which can deal efficiently with multicollinearity of input features); (c) the addition of new features that were found to be very relevant; (d) the proposal of an explicit feature selection method to identify relevant features in a systematic way; and (e) the exploitation of multiple datasets of translations from different MT systems and language pairs, with different types of human and automatic quality annotations, through the use of resource-independent features and the definition of system-independent features.

3 Experimental setting

3.1 Features

We extract all the features identified in previous work for sentence-level CE (see (Blatz et al., 2003) for a list), except those depending on linguistic resources like parsers or WordNet. We also add new features to cover aspects that have not been directly addressed in previous work, including the mismatch of many superficial constructions between the input and output sentences (percentages of punctuation symbols, numbers, etc.), similarity between the source sentence and sentences in a monolingual corpus, word alignment between input and output sentences, length of phrases, etc. This results in a total of 84 features.

Many of these features depend on some aspect of the translation process, and therefore are MT system-dependent and could not be extracted from all translation data used in this paper. We thus divide the features in two subsets: (a) *black-box features*, which can be extracted given only the input sentence and the translation produced by the MT system, i.e., the source and target sentences, and possibly monolingual or parallel corpora, and (b) *glass-box features*, which may also depend on some aspect of the translation process.

The black-box group includes simple features like source and target sentence lengths and their ratios, source and target sentence n-gram frequency statistics in the corpus, etc. This constitutes an interesting scenario and can be particularly useful when it is not possible to have access to internal features of the MT systems (in commercial systems, e.g.). It also provides a way to perform the task of CE across different MT systems, which may use different frameworks. An interesting research question is whether it is possible to produce accurate CE models taking into account only these very basic features. To our knowledge, this issue has not been investigated before.

The glass-box group includes internal features of the MT system, like the SMT model score, phrase and word probabilities, and alternative translations per source word. They also include features based on the n-best list of translation candidates, some of which apply globally to the set of all candidates for a given source sentence (e.g. degree to which phrases are translated in the same way throughout the n-best list), and some to specific candidates (e.g. ratio between scores of the candidate and top candidate). We extract a total of 54 glass-box features.

3.2 Data

We use two types of translation data: (a) translations automatically annotated with NIST scores, and (b) translations produced by different MT systems and for multiple language-pairs, manually annotated with different types of scores.

The automatically annotated dataset, henceforth *NIST dataset*, is produced from the French-English Europarl parallel corpus, as provided by the WMT-2008 shared translation task (Callison-Burch et al., 2008). We translate the three development-test sets available ($\sim$6k sentences) using a phrase-based MT system [omitted for blind review]. These translations and their 1,000 n-best lists are scored according to sentence-level NIST and the 84 features are extracted from them.

The dataset is first sampled into 1,000 subsamples, where each subsample contains all feature vectors for a certain position in all the n-best lists and is randomly split in training (50%), validation (30%) and test (20%) using a uniform distribution.

The first type of manually annotated datasets (*WMT datasets*) is derived from several corpora of the WMT-2006 translation shared task (Koehn and Monz, 2006). These are subsets of sentences from the test data used in the shared task, annotated by humans according to adequacy, with scores from 1 (worst) to 5 (best). Each corpus contains $\sim$100-400 sentences and refers to a given language pair and MT system. Since this number is very small, we put together all sets of translations from a given MT system. We select four among the resulting datasets: the three phrase-based SMT systems (*S1, S2, S3*) with the high-

est numbers of examples and the only rule-based system (*RB*). Each new dataset contains ∼1,300-2,000 sentences, and 4-6 language-pairs. The feature vectors of these datasets contain only black-box features. To account for mixing language-pairs, we add the source and target language indicators as features. The task becomes predicting the quality of a given MT system which translates between different language pairs.

The manually annotated datasets of the second type (*1-4 datasets*) contain 4K sentences of the Europarl domain (English-Spanish), translated by four SMT systems developed by different partners in the project *P* [omitted for blind review]: *P-ES-1, P-ES-2, P-ES-3 and P-ES-4*. The sentences are annotated by professional translators according to 1-4 quality scores, which are commonly used by them to indicate the quality of translations with respect to the need of post-edition: 1 = requires complete retranslation, ..., 4 = fit for purpose.

Datasets of the final type (*post-edition datasets*) contain 3K sentences of the automotive industry domain (English-Russian), translated by three MT systems from the same project *P*: *P-ER-1, P-ER-2 and P-ER-3*. The sentences are annotated according to post-edition time, that is, given a source sentence in English and its translation into Russian, a professional translator post-edited such translation to make it into a good quality sentence, while the time was recorded.

Black-box features are extracted from all datasets in the last two groups (*1-4* and *post-edition*). Additionally, glass-box features are extracted from one of the datasets (*P-ES-1*), since we had access to the SMT system in this case. We call this *P-ES-1gb*. In the *post-edition* datasets, the post-edition time is first normalized by the source sentence length, so that the score refers to the time necessary per source word.

For each manually annotated dataset, the feature vectors are randomly subsampled 100 times in training (50%), validation (30%) and test (20% using a uniform distribution.

In both automatically and manually annotated datasets, we represent each subsample as a matrix of variable predictors (X) times variable response (Y) and normalize feature values using the *z* score.

Datasets covering different language pairs and MT systems and particularly data annotated according to post-edition time for CE have not been investigated before.

3.3 Learning algorithm

We estimate the quality of the translations by predicting the sentence-level NIST, 1-5 / 1-4 scores or post-edition time using Partial Least Squares (PLS) (Wold et al., 1984). Given a matrix X (input variables) and a vector Y (response variable), the goal of PLS regression is to predict Y from X and to describe their common structure. In order to do that, PLS projects the original data onto a different space of latent variables (or "components") and is also able to provide information on the importance of individual features in X. PLS is particularly indicated when the features in X are strongly correlated (multicollinearity). This is the case in our datasets. For example, we consider each of the SMT system features individually, as well as the sum of the all these features (the actual SMT model score). With such datasets, standard regression techniques usually fail (Rosipal and Trejo, 2001). PLS has been widely used to extract qualitative information from different types of data (Frenich et al., 1995), but to our knowledge, it has not been used in NLP applications. More formally, PLS can be defined as an ordinary multiple regression problem, i.e.,

$$Y = XB_w + F$$

where B_w is the regression matrix, F is the residual matrix, but B_w is computed directly using an optimal number of components. For more details see (Jong, 1993). When X is standardized, an element of B_w with large absolute value indicates an important X-variable.

It is well known that feature selection can be helpful to many tasks in NLP, and that even learning methods that implicitly perform some form of feature selection, such as SVMs, can benefit from the use of explicit feature selection techniques. We take advantage of a property of PLS, which is the ordering of the features of X in B_w according to their relevance, to define a method to select subsets of discriminative features (Section 4).

To evaluate the performance of the approach, we compute the average error in the estimation of NIST or manual scores by means of the Root Mean Squared Prediction Error (RMSPE) metric: $\sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - \widehat{y}_j)^2}$, where N is the number of points, $\widehat{y}$ is the prediction obtained by the regressor and y is the actual value of the test case. RMSPE quantifies the amount by which the estimator differs from the expected score: the lower the value,

the better the performance.

4 Method

Our method to perform regression supported by an embedded feature selection procedure consists of the following steps: (1) sort all features according to their relevance in the training data; (2) select only the top features according to their relevance in the validation set; (3) apply the selected features to the test data and evaluate the performance. In more details:

1. Given each pre-defined number of components, for each *i-th* subsample of the training data, we run PLS to compute the $B_w(i)$ matrix, generating a list $L_b(i)$ of feature ranked in decreasing order of importance. After generating L_b for all subsamples, we obtain a matrix where each row i contains an $L_b(i)$, e.g.:

66	7	56	...	10
44	56	3	...	10
...	...	...	...	...
66	56	3	...	10

   A list L containing the global feature ordering for all subsamples is obtained by selecting the feature appearing most frequently in each column (i.e., taking the *mode*, without repeating features). In the case shown, $L = \{66, 56, 3, \ldots, 10\}$.
2. Given the list L produced for a certain number of components, for each *i-th* subsample of the validation data, we train the regression algorithms on 80% of the data, adding features from L one by one. We test the models on the remaining validation data and plot the learning curves with the mean error scores over all the subsamples. By analyzing the learning curves, we select the first n features that maximize the performance of the models.
3. Given the selected n features and the number of components that optmized the performance in the validation data, for each *i-th* subsample of the test data, we train (80%) and test (20%) the performance of the regressor using these features, and compute their corresponding metrics over all subsamples.

5 Results

5.1 NIST dataset

Figure 1 illustrates the performance for different numbers of PLS components used to generate ordered lists of features. The maximum performance

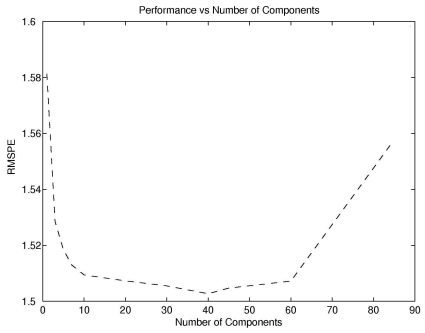


Figure 1: Performance for lists generated with different numbers of components - *NIST dataset*

is obtained from the ordered list generated with 40 components. This resulted in 32 features being selected, and an RMSPE in the test set of **1.503** $\pm$ **0.045**. The RMSPE for all features, without applying the feature selection method, is **1.670** $\pm$ **0.669**. Therefore, the models produced for the selected subset of features perform better than using all features. Moreover, results for the subsets of features are more stable, given the large variance observed in the RMSPE score with all features. To provide a more intuitive measure, we can say that the system deviates on average ~ 1.5 points when predicting the sentence-level NIST score. We believe this is an acceptable deviation, given that the scores vary from 0 to 18.44.

Although the subsets of features selected vary for different numbers of components, some appear in all the top lists:

- average number of alternative translations for words in the source sentence;
- ratio of source and target lengths;
- proportion of aborted nodes in the decoder's search graph.

The first feature reflects the ambiguity and therefore the difficulty of translating the source sentence. The second favors source and target sentences which are similar in size, which is expected for close language-pairs like English-French. The last gives an idea about the uncertainty in the search: nodes are aborted if the decoder is certain that they will not yield good translations.

Other features appear as relevant for most choices in the number of components:

- source sentence length;
- number of different words in the n-best list divided by average sentence length;

MT	RMSPE	RMSPE all features
RB	1.058 ± 0.087	1.171 ± 0.098
S1	1.159 ± 0.064	1.197 ± 0.059
S2	1.116 ± 0.073	1.190 ± 0.073
S3	1.160 ± 0.059	1.201 ± 0.062

Table 1: RMSPE - *WMT datasets*

- 1/3-gram source frequency statistics in the whole corpus or its most frequent quartile;
- 3-gram source language model probability;
- 3-gram target language model probability considering n-best list as corpus;
- phrase probabilities;
- average size of phrases in the target sentence;
- proportion of pruned and remaining nodes in decoder's final search graph.

These features in general point out the difficulty of translating the source sentence, the uniformity of the candidates in the n-best list, how well the source sentence is covered in the training corpus, and how commonplace the target sentence is. They include some SMT model features, but notably not the actual SMT score. Surprisingly, half of these very discriminative features are black-box.

5.2 Manually annotated datasets

Results for the *WMT datasets* are less straightforward to interpret, since the problem has more variables, particularly multiple language pairs, in- / out-of-domain sentences in a single dataset, and reduced dataset sizes. The best numbers of components vary from 1 to 25 and feature selection results in different subsets of features (from 2 to 10 features) for different MT systems. Nevertheless, in all the datasets, feature selection yields better results, as shown in Table 1.

The models deviate on average ~ 1.1 points when predicting 1-5 scores. This means, e.g., that some sentences atually scoring 4 would be given to the user as scoring 5.

Table 2 shows the performance obtained for the *1-4* and *post-edition datasets*. The figures for the subsets of features consistently outperform those for using all features and are also more stable.

The models produced for different MT systems (P-ES-1 to P-ES-4) deviate ~ 0.6-0.7 points when predicting the sentence-level 1-4 scores, which we believe is a satisfactory deviation. For example, one sentence that should be considered as "fit for purpose" (score 4) would never be predicted as "requires complete retranslation" (score 1) and discarded as a consequence.

MT	RMSPE	RMSPE all features
P-ES-1gb	0.690 ± 0.052	0.780 ± 0.385
P-ES-1	0.706 ± 0.059	0.793 ± 0.643
P-ES-2	0.653 ± 0.114	0.750 ± 0.541
P-ES-3	0.718 ± 0.144	0.745 ± 0.287
P-ES-4	0.603 ± 0.262	1.550 ± 3.551
P-ER-1	1.951 ± 0.174	2.083 ± 0.561
P-ER-2	2.883 ± 0.301	3.483 ± 1.489
P-ER-3	3.879 ± 0.339	4.893 ± 2.342

Table 2: RMSPE - *1-4* and *post-edition datasets*

An interesting result is the comparison between the scores for the two variations of the first dataset, i.e., *P-ES-1gb* (glass-box features) and *P-ES-1* (black-box features). The gain in using glass-box features is very little in this case. This shows that although glass-box features may be very informative, it is possible to represent the same information using simpler features. From a practical point of view, this is very important, since black-box features are usually faster to extract and can be used with any MT system.

In order to investigate whether any single feature would be able to predict the quality scores as well as the combination of selected good features, we compare the Pearson's correlation coefficient of each feature and the predicted CE score with the expected human score. The correlation of the best features with the human score is ~ 0.5 (glass-box features) or up to ~ 0.4 (black-box features) across the different *1-4 datasets*. The CE score correlates ~ 0.6 with the human score.

In Table 3 we compare the correlation of the CE and human scores against that of well-known MT evaluation metrics (at the sentence level) and human scores on a test set for *P-ES-1gb* (values are similar for other datasets). The quality estimate predicted by our method correlates better with human scores than reference-based MT evaluation metrics. We apply bootstrapping re-sampling on the data and then use paired t-test to determine the statistical significance of the correlation differences (Koehn, 2004). The differences between all metrics and CE are statistically significant with 99.8% confidence. Different from these metrics, our method requires some training data for a given language-pair and text domain, but once ths training is done, it can be used to estimate the quality of any number of new sentences.

Results for the *post-edition* datasets vary considerably from system to system. This may indicate that different MT systems require more post-

BLEU-2	NIST	TER	Meteor	CE score
0.342	0.298	-0.263	0.376	0.602

Table 3: Correlation of MT evaluation metrics and our score with human annotation - *P-ES-1gb*

edition due to their translation quality. For example, taking the error for *P-ER-1*, of ~ 1.95, we can say that the CE system is able to predict, for a given source sentence, a post-edition time by source word that will deviate up to 1.95 seconds from the real post-edition time needed. The average errors found may seem a very large on a word-basis, but more investigation on the use of this type of CE score to aid translators in their post-edition work is necessary in this direction.

By analyzing the top features in all tasks with the manually annotated datasets we can highlight the following ones:

- source language and in/out-of-domain indicators (*WMT datasets*);
- source & target sentence 3-gram language model probability;
- source & target sentence lengths;
- percentages of types of word alignments;
- percentage and mismatch in the numbers and punctuation symbols in the source and target.

The first two features convey corpus information. Their impact in the performance is expected, given that it may be easier to translate between certain pairs of languages and in-domain sentences. The size of the source and target points out the difficulty of the translation (longer sentences are more difficult). Like the remaining features, it also expresses some form between source and target.

6 Discussion and conclusions

We have presented a series of experiments on a method for confidence estimation to MT that allows taking into account the contribution of different features and have also identified very informative and non-redundant features that improve the performance of the produced CE models. Although it is not directly possible to compare our results to previous work, because of the unavailability of the datasets used before, we consider our results to be satisfactory. Particularly in the case of the regression task, it is possible to have some intuition on what the impact of the error would be. For example, it would indicate crossing on average one category in the quality ranking of the tasks predicting adequacy scores (1 = worst, 5 = best), and only result in uncertainty in the boundaries between two adjacent categories in the *1-4 datasets*.

The sets of relevant features identified includes many features that have not been used before, including the average size of the phrases in the target, several types of mismatchings in the source and target, etc. Some of the others features have been used in previous work, but their exact definition is different here. For example, we use the *proportion* of aborted search nodes, instead of absolute values, and we compute the average number of alternative translations by using probabilistic dictionaries produced from word-alignment.

Besides directly using the estimated scores as quality indicators to professional translators or end-users, we plan to further investigate uses for the features selected across MT systems and language pairs from different MT points of view. In the experiments with the *NIST dataset*, the features found to be the most relevant are not those usually considered in SMT models. Simple features like the ratio of lengths of source and target sentences, the ambiguity of the source words, the coverage of the source sentence in the corpus are clearly good indicators of translation quality. A future direction will be to investigate whether these features could also be useful to improve the translations produced by SMT systems, e.g., in the following ways:

- Complement existing features in SMT models.
- Rerank n-best lists produced by SMT systems, which could make use of the features that are not local to single hipotheses.

As discussed in (Gamon et al., 2005), the readability of the sentence, expressed by features like 3-gram language models, is a good proxy to predict translation quality, even in terms of adequacy. Ultimately, automatic metrics such as NIST aim at simulating how humans evaluate translations. In that sense, the findings of our experiments with the manually annotated datasets could also be exploited from an MT evaluation point of view, for example, in the following ways:

- Provide additional features to a reference-based metric like that proposed by (Albrecht and Hwa, 2007a).
- Provide a score to be combined with other MT evaluation metrics using frameworks like

those proposed by (Paul et al., 2007) and (Giménez and Màrquez, 2008).

Our findings could also be used to provide a new evaluation metric on itself, with some function to optimize the correlation with human annotations, without the need of reference translations.

References

Albrecht, J. and R. Hwa. 2007a. A re-examination of machine learning approaches for sentence-level mt evaluation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 880–887, Prague.

Albrecht, J. and R. Hwa. 2007b. Regression for sentence-level mt evaluation with pseudo references. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 296–303, Prague.

Blatz, J., E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2003. Confidence estimation for machine translation. Technical report, Johns Hopkins University, Baltimore.

Blatz, J., E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th Conference on Computational Linguistics*, pages 315–321, Geneva.

Callison-Burch, C., C. Fordyce, P. Koehn, C. Monz, and J. Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the 3rd Workshop on Statistical Machine Translation*, pages 70–106, Columbus.

Doddington, G. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the 2nd Conference on Human Language Technology Research*, pages 138–145, San Diego.

Frenich, A. G., A. G. Jouan-Rimbaud, D. Massart, D. L. Kuttatharmmakul, S. Martinez Galera, and J. L. M. Martinez Vidal. 1995. Wavelength selection method for multicomponent spectrophotometric determinations using partial least squares. *Analist*, 120(12):2787–2792.

Gamon, M., A. Aue, and M. Smets. 2005. Sentence-level mt evaluation without reference translations: beyond language modeling. In *Proceedings of the European Association for Machine Translation Conference*, Budapest.

Gandrabur, S. and G. Foster. 2003. Confidence estimation for translation prediction. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 95–102, Edmonton.

Giménez, J. and L. Màrquez. 2008. Heterogeneous automatic mt evaluation through non-parametric metric combinations. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, pages 319–326, Hyderabad.

Jong, S De. 1993. Simpls: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263.

Kadri, Y. and J. Y. Nie. 2006. Improving query translation with confidence estimation for cross language information retrieval. In *Proceedings of the 15th Conference on Information and Knowledge Management*, pages 818–819, Arlington.

Koehn, P. and C. Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 102–121, New York.

Koehn, P. 2004. Statistical significance tests for machine translation evaluation. In *Conference on Empirical Methods in Natural Language Processing*, Barcelona.

Kulesza, A. and A. 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.

Papineni, K., S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, page 311318, Morristown.

Paul, M., A. Finch, and E. Sumita. 2007. Reducing human assessment of machine translation quality to binary classifiers. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation*, pages 154–162, Skovde.

Quirk, C. B. 2004. Training a sentence-level machine translation confidence measure. In *Proceedings of the 4th Conference on Language Resources and Evaluation*, pages 825–828, Lisbon.

Rosipal, R. and L. J. Trejo. 2001. Kernel partial least squares regression in reproducing kernel hilbert space. *Machine Learning Research*, 2:97–123.

Ueffing, N. and H. Ney. 2005. Application of word-level confidence measures in interactive statistical machine translation. In *Proceedings of the 10th Conference of the European Association for Machine Translation*, pages 262–270, Budapest.

Wold, S., A. Ruhe, H. Wold, and W. J. Dunn. 1984. The covariance problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific Computing*, 5:735–743.

Evaluation-guided pre-editing of source text: improving MT-tractability of light verb constructions

Bogdan Babych **Anthony Hartley** **Serge Sharoff**
Centre for Translation Studies
University of Leeds, LS2 9JT, UK
{b.babych,s.sharoff,a.hartley}@leeds.ac.uk

Abstract

This paper reports an experiment on evaluating and improving MT quality of light-verb construction (LVCs) – combinations of a 'semantically depleted' verb and its complement. Our method uses construction-level human evaluation for systematic discovery of mistranslated contexts and creating automatic pre-editing rules, which make the constructions more tractable for Rule-Based Machine Translation (RBMT) systems. For rewritten phrases we achieve about 40% reduction in the number of incomprehensible translations into English from both French and Russian. The proposed method can be used for enhancing automatic pre-editing functionality of state-of-the-art MT systems. It will allow MT users to create their own rewriting rules for frequently mistranslated constructions and contexts, going beyond existing systems' capabilities offered by user dictionaries and do-not translate lists.

1 Introduction: Automatic rewriting functionality for MT systems

Current state-of-the-art RBMT systems offer users customisable functionality for the transfer stage, in the form of user-definable do-not-translate lists and user dictionaries. However, the source language analysis capabilities of MT systems remain largely inaccessible to users and such systems still do not offer any support for rewriting at the pre-editing stage. Rewriting offers a way of enhancing the comprehensibility of MT output by more efficiently exploiting existing transfer resources of an MT engine, and greatly extends coverage of construction in a customised way without any changes to the engine.

In our paper we suggest that users will benefit from an integration of source-language rewriting capabilities into MT systems, and from their synergies with existing tools – user dictionaries and do-not-translate lists. Such integration will offer users much greater flexibility, because certain phenomena are much better treated in a monolingual rewriting stage rather than within the dictionary. For example, rewriting can 'repair' intractable word order, or handle discontinuous multiword expressions in a much more principled way than a user dictionary.

We describe an experiment which demonstrates the usefulness of source-rewriting functionality for state-of-the-art MT systems. For our evaluation-guided rewriting experiment we have chosen light-verb constructions (LVCs) – verb phrases consisting of a 'light', i.e., 'semantically depleted' verb and its object. Objects in such constructions are so-called logical predicates – such as names of actions, activities, states, properties, relations – that put forward some situational propositions. The relationship between verbs and complements can be described in terms of lexical functions like $Oper_1$ (Mel'čuk, 1996). LVCs are very common in a variety of languages, e.g., *take action, take part, put pressure, make a decision*} in English, *оказывать давление 'put pressure'* (lit. 'make pressure') in Russian, or *tenir compte de 'take into account'* (lit. 'hold account of') in French (Salkoff, 1999).

These constructions are often mistranslated by state-of-the-art MT systems, since they often require non-literal translation (En: *take part* > Ru: *~'accept part'*). Many of these constructions have synonymous verbs or other phrases, which can facilitate MT: (*take part --> participate*). We used such near-synonyms for rewriting problematic LVCs and evaluated the effect of this rewriting on the comprehensibility of the translations.

For this purpose we selected a group of LVCs for several frequent French and Russian light verbs and assessed the comprehensibility of their English translations. An initial analysis of the comprehensibility of LVCs aimed to identify the

Proceedings of the 13th Annual Conference of the EAMT, pages 36–43,
Barcelona, May 2009

most problematic constructions, whose re-writing could make the biggest impact on MT output quality.

RBMT systems generally have knowledge of the most frequent LVCs. For instance, ProMT does handle constructions with *принимать решение* reasonably, translating it as *make a decision*, not as *accept a decision*. Systran sensibly translates French LVCs like *commettre un crime* as *commit a crime*, and *commettre une erreur* as *make an error*. However, the coverage is not consistent. For instance, for the same directions Systran does not handle *faire en sorte* (*'do in such a way as to'*) in French, while ProMT does not handle *брать штурмом* (*'make an assault'*) in Russian. Hence, a feasible alternative is to rewrite problematic constructions in the source text to produce input that can be more sensibly handled by an individual MT system.

An insight into 'post-editing the source text' is offered by Somers (1997). For instance, he suggested re-writing source English country names (ungrammatically) as *the France*, *the Japan* for translation into French to improve the grammaticality of the output (*la France, le Japon*), where an MT system fails to insert the required definite article.

Our experiment evaluates improvements in MT quality, which can be achieved for a particular class of linguistic constructions, if such automatic pre-processing mechanisms are systematically implemented for state-of-the-art RBMT systems.

2 Previous work

Lexical and structural ambiguities in natural language have been traditionally fundamental problems in MT, which seriously limit the quality of its output, especially on unrestricted input text. However, if expressions in the source text are restricted in certain ways, the performance of MT systems can be improved considerably. For general-purpose MT systems, this observation has led to recommendations and tools for MT-oriented authoring and pre-editing of source texts –– so called MTranslability (Bernth and Gdaniec, 2001). For specific technical domains – such as software, aviation or automotive documentation – this observation has led to successful applications of Controlled Language (CL), which minimises post-translation editing (Nyberg et al., 2003) and usually works in conjunction with MT engines customised to match CL specifications.

A disadvantage of this approach is that very few texts are written using CL, so such recommendations are not directly applicable to the majority of texts that have to be translated for assimilation purposes using MT.

The proposed automatic rewriting mechanisms for MT will enable the users to accurately preprocess complex contexts of constructions from general language, which are commonly mistranslated by MT systems.

The remaining sections are organised as follows: In Section 3 we present the design of our experiment for evaluating the improvement achieved by modifying LVCs. In Section 4 we discuss its results and implications for creating re-writing rules for making MT input more tractable. Finally, in Section 5 we discuss other ideas possible in this field.

3 Method

3.1 Identifying LVCs

In our experiment we took the point of view of users of modern commercial MT systems (typically – medium or large translation companies), who want to improve comprehensibility of MT output, focussing on the contexts of particular classes of linguistic constructions, in our case we selected light-verb constructions (LVCs).

Our evaluation-guided procedure for systematically discovering frequent mistranslated contexts of such constructions and designing automatic rules to change them into a more MT-tractable form can be carried out by such users, who would find automatic rewriting mechanism for the pre-processing stage very useful. The procedure consists of the following stages.

As a first stage we generated lists of noun collocations for seven French and eight Russian light verbs, based on a study of LVCs by (Mudraya et al. 2008): French *commettre, donner, faire, mettre, passer, prendre, rendre* and Russian *брать (take), вести (lead), давать (give), делать (make), иметь (have), нести (carry), положить (lay), ставить (put)*. We manually checked the top-ranking collocates sorted by log-likelihood association scores and selected 63 French and 55 Russian LVCs (e.g., *принимать закон/меры/решение* ('pass a law', 'take measures', 'make a decision'). For each LVC we generated concordance lines (in a window of about 20 words) from French and Russian Internet corpora and translated them using three MT engines: French>English Systran 5.0,

Russian>English Systran 5.0 and Russian>English ProMT 8.0.

We randomly selected and analysed up to 25 concordance lines for each construction (the selection was intentionally small to model the real-world scenario of evaluation-guided improvement of MT for potential industrial users of the technology), and we identified those LVCs with the least intelligible translations, e.g.:

(3) Ne **faisons** pas **confiance** aux Anglais.
-> Systran: Let us not make confidence with the English.
Automatically rewritten ST:
Ne **comptons** pas **sur** les Anglais.
-> Systran: Let us not count on the English.

Since for Russian two MT systems were available, we used one of them (Systran) for identifying problematic LVCs, and the other (ProMT) for a 'blind rewriting' experiment, where rules and constructions selected for Systran were also applied to ProMT translation. The performance of ProMT on such constructions was, then, not known in advance.

Since MT systems can differ in their coverage of problematic constructions, this experiment was designed to assess to what extent the set of rewriting rules is system-dependent. Equally, it sought to establish whether re-writing rules are 'portable' from one system to another, that is, whether certain classes of language constructions are generally intractable for RBMT.

3.2 Rewriting of LVCs

The comprehensibility of certain LVCs clearly benefits from the rewriting of the source. Overall, nine of the LVCs identified for French exhibited this type of problem for at least some of their contexts of occurrence (*faire appel, faire confiance, faire face, faire (en) sorte, donner lieu, donner rendez-vous, mettre fin, prendre conscience, metre (la) main*). These constructions were selected for rewriting and we created rewriting rules for all their problematic contexts. Modifications mostly involved replacing the verb and keeping the noun (as the central meaning component) or replacing the whole construction. For example, the rewriting table for *faire confiance* is presented in Table 2:

Note that in our experiment separate rules were created for each combination of word forms that occurred in concordances, which was supposed only to simulate capabilities of real rewriting mechanisms that can be developed for state-of-the-art commercial MT systems. These rules in practice should be written in a more general way, since the rewriting system can have access to lexical and morphological features of word forms developed for the translation engines, e.g., then the last 3 rules in Table 2 could be merged into a single rule:

(3) [lemma='faire'] pas confiance [lemma=à] [lemma='le']?

The re-writing procedure was applied universally to all examples of LVCs, even if some examples were understandable in their original form.

faisaient au moins confiance à --> comptaient au moins sur
faire davantage confiance à --> compter davantage sur
fais totalement confiance a --> compte totalement sur
fais pas confiance à --> compte pas sur
faisaient pas confiance au --> comptaient pas sur le
faisons pas confiance aux --> comptons pas sur les […]

Table 2. Rewriting table for *faire confiance*

3.3 Evaluation of baseline translation quality for Russian LVCs

For Russian>English Systran 5.0 translations the problems with LVCs were more serious, so we carried out a systematic evaluation of contextual comprehensibility for all 55 LVCs. The comprehensibility of each MT-translated concordance line was annotated on 1-3 scale:

"3" - high confidence I understand correctly
"2" - low confidence I understand correctly
"1" - do not understand at all

The score was given to each concordance line out of 25 randomly selected contexts, and average scores were computed for each source-language LVC that generated these translations. Then these LVCs were ranked by their average scores, and the lowest ranking constructions were identified as those which need to be rewritten. Table 3 shows the numbers of Russian LVCs in the different ranges of comprehensibility scores. In all, 19 LVCs from the three lowest

groups of LVCs were selected for rewriting, since their average scores are centred around 'low confidence' or 'incomprehensible' scores.

Score range	Number of LVCs
[2.6 … 3.0]	23
[2.2 < 2.6)	13
[1.8 < 2.2)	10
[1.4 < 1.8)	5
[1.0 < 1.4)	4

Table 3. Comprehensibility of Russian>English LVCs

This analysis illustrates the extent of the LVC problem for the Russian>English MT system: 19 of 55 frequent LVCs (35%) generate low confidence translations and 9 of them (16%) produce mostly incomprehensible MT output.

Human evaluation scores before rewriting LVCs, and the extent to which LVC rewriting improves these figures are negatively correlated, so it is harder to achieve improvement by rewriting more comprehensible contexts. In particular, Pearson's correlation coefficient *r* between the baseline quality of LVC translations and the extent to which the quality could be improved via rewriting for individual LVCs is -0.71, and for averages for the ranked groups of six LVCs it becomes -0.99.

Therefore, we chose to rewrite the 19 LVCs with lowest evaluation scores, which should clearly benefit from rewriting.

3.4 Evaluators and evaluation packs

The results of re-writing were tested in an evaluation experiment. The comprehensibility of LVCs was judged by 16 native English speakers (Masters students in translation), who did not see the source text. The judges completed a questionnaire like that shown in Figure 1.

Evaluators judged concordances for LVCs in 10 different evaluation packs. Each evaluation pack contained 47 pairs of contexts for comparison: exactly one pair of randomly selected contexts for each LVC came from each of the three MT engines: French>English and Russian>English Systran 5.0 and another state-of-the-art Russian>English MT system ProMT 8.0. This system was used for 'blind' rewriting: the baseline performance of ProMT 8.0 on LVCs was unknown to us, and rewriting was done exactly as for Systran, without any preliminary system-specific tuning. The order within each pair of LVC contexts – left/right *vs* before/after rewriting – was also randomised, so evaluators did not know which context was the baseline, and which was experimental.

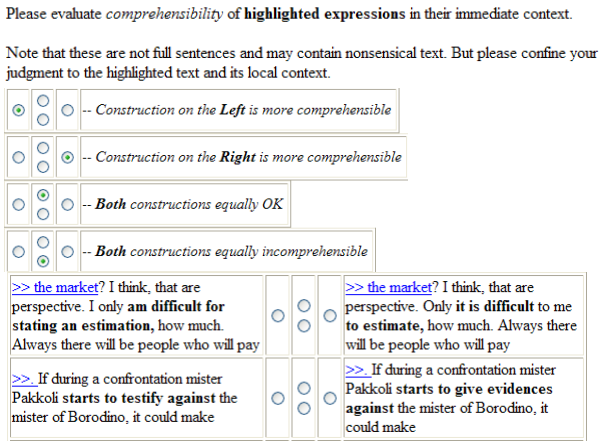


Figure 1. Evaluation questionnaire

There were two independent judgements for each context in the first six evaluation packs, and one judgement for contexts in the remaining packs. Evaluators gave 700 independent comparison judgements in total: 300 for each Russian>English system and 100 for French>English Systran.

Evaluation scores were converted to a numeric scale as shown in Table 4:

	Score Before RW	Score After RW
Before RW more comprehensible	+1	–1
After RW more comprehensible	–1	+1
Both equally comprehensible	+1	+1
Both equally incomprehensible	–1	–1

Table 4. Numeric conversion of evaluation scores

Numeric values were used for computing average scores for evaluators, MT systems and contexts of the same LVC, and for measuring the degree of improvement in these cases.

In our experiment average inter-annotator agreement measured by Cohen's kappa coefficient was around 0.28, which is a typical figure for human MT evaluation (Ye at al., 2007: 242). Still, our experiment was different from traditional MT evaluation, because human judges did not see complete sentences. We specifically asked our evaluators to confine their judgements to highlighted LVCs and their local context.

4 Results

4.1 Overall system evaluation

Chart 3 shows the overall number of comprehensible / incomprehensible translations before and after rewriting for Russian>English Systran 5.0.

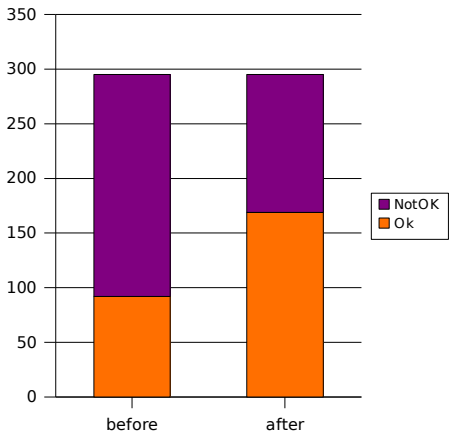


Chart 3. Results of LVC rewriting: Ru>En Systran5

The chart shows that rewriting of problematic contexts for this MT system gives a 38% reduction in incomprehensible translations and an 84% increase in comprehensible translations. French>English Systran showed even better improvement, while for Russian>English ProMT, which used blind rewriting, the improvement was smaller. Tables 5 and 6 summarise these improvement figures. Table 5 represents average evaluation scores on the [-1...+1] comprehensibility scale and proportional change in the number of comprehensible / incomprehensible contexts after rewriting.

	Before	**After**	**InC**	**C**
fr>en-Systran	-0.73	0.02	-44%	+283 %
ru>en-Systran	-0.38	0.15	-38%	+ 84 %
ru>en-ProMT*	-0.37	-0.03	-25%	+ 54 %

Before: average score before rewriting
After: average score after rewriting
InC: change in incomprehensible
C: change in comprehensible

Table 5. Average evaluation scores and changes in number of contexts

Note that for both Systran engines (Russian>English and French>English) evaluation-guided rewriting brought average scores above zero, which can be viewed as a comprehensibility threshold. Blind rewriting for Russian>English (ru>en-promt*) also brought an increase in average scores, but not enough to cross the threshold.

Table 6 shows proportions of scores in each category for the 3 MT engines.

	bothX	**before+**	**after+**	**both+**
fr-en-systr	0.42	**0.07**	**0.44**	0.07
ru-en-systr	0.29	**0.14**	**0.40**	0.18
ru-en-promt*	0.29	**0.23**	**0.40**	0.09

bothX: both Not OK
before+: before rewriting more comprehensible
after+: after rewriting more comprehensible
both+: both OK

Table 6. Improvement across MT systems

Again, for blind rewriting there is a much greater proportion of contexts which were judged as being 'better before rewriting'.

4.2 Construction-level evaluation

An evaluation of individual constructions provides a finer-grained analysis of the effect of LVC rewriting on comprehensibility. Charts 4 and 5 represent average scores before and after evaluation-guided rewriting, for each of French and Russian LVCs translated by Systran.

It can be seen from these charts that evaluation-guided rewriting normally increases comprehensibility of LVC contexts. Only 11% to 16% of LVCs show slight degradation in comprehensibility or no change.

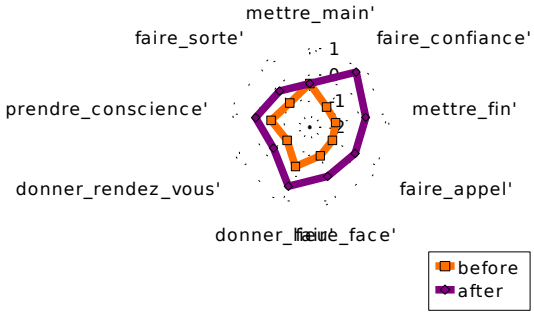


Chart 4. Average scores for Fr LVCs; Fr>En Systran 5.0

However, for blind rewriting 37% of LVCs showed a decline or no change in comprehensibility. Chart 6 illustrates these results. On this chart there is an area where the 'before rewriting' line is outside from the 'after rewriting' line, meaning that there is degradation of comprehens-

ibility for some constructions. Note that the majority of LVCs in this group are highly comprehensible before rewriting, and in this case rewriting decreases comprehensibility.

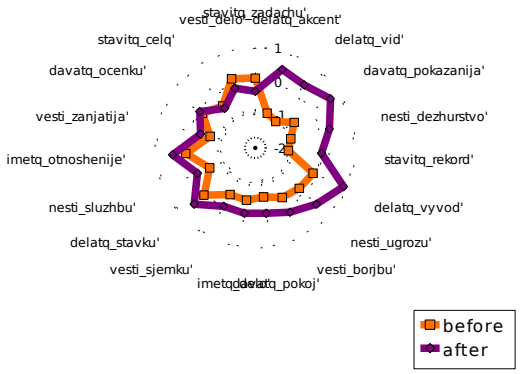


Chart 5. Average scores for Ru LVCs; Ru>En Systran 5.0

Surprisingly, two LVCs showed slight deterioration after evaluation-guided rewriting, but high improvement after blind rewriting: *ставить задачу (set a task): -0.4 vs +0.7* and *ставить цель (set a goal*): *-0.1 vs +1.11*, and only one Russian LVC showed no improvement at all for both engines: *вести дело (carry out business):* average scores changed by *-0.3* and *-0.2*

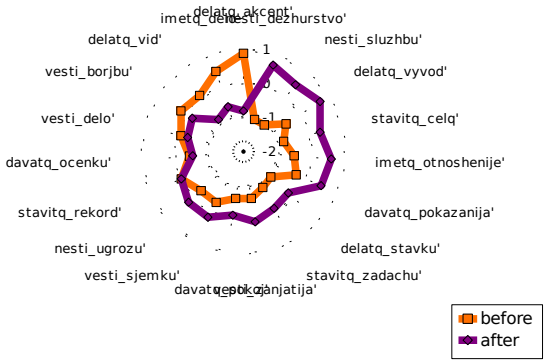


Chart 6. Blind rewriting: Average scores for Ru LVCs; Ru>En ProMT 8.0

5 Discussion

5.1 Automatic rewriting capabilities of MT systems

Our experiment demonstrated that the comprehensibility of translations of certain types of constructions, such as LVCs, can be improved via automatic rewriting of the source text into synonymous and more MT-tractable constructions.

Rewriting gives users gentle control over the analysis stage in the MT architecture, so they do not need to rely excessively on user dictionaries, which intervene at the transfer stage.

A general architectural consensus for RB MT is to try to do as much work as possible at the analysis stage and to keep transfer simple. Rewriting functionality will allow users to classify some problems, such as incomprehensible LVCs, as analysis problems, and deal with them in a more principled way using the existing capabilities of their MT engine, before such constructions are adequately covered by a new release of the engine. Automatic rewriting can also be more cost-efficient, e.g., for organisations which use MT for translating into several languages.

A rewriting mechanism can be more efficient in relying on the source-language analysis modules of the MT engine for extracting morphological and lexical features, multiword expressions, etc., and in allowing users to write rules in a compact way.

Traditionally user dictionaries do not handle discontinuous multiword expressions, even within the top commercial MT systems. Automatic pre-processing functionality can fill this gap, allowing the users to correct more MT problems than they currently handle via updating user dictionaries, e.g.:

(1) Законодательная власть принимает сотни законов ежегодно.
(Literal translation: Legislative power **passes** hundreds of **laws** yearly)
→ Systran: Legislative authority assumes hundreds of laws yearly.

The Russian construction *принимать закон (to pass a law)* has been mistranslated by Systran, so (1) is incomprehensible unless the reader knows Russian. Currently the entries in Systran's user dictionaries fail to match constructions with a gap, e.g., ***vote*** *hundreds* ***of laws***.

Adding a pair "*принимать ~ to pass"* into the MT dictionary will be detrimental for other contexts, because translation of this Russian verb, even in the same domain, can be very different: *to accept, to admit, to join, to take, to pass…*

Automatic rewriting rules on the design stage can take into account the context of rewritten constructions, and consequently – efficiently cover discontinuous multiword expressions. In the previous example when the verb of the source Russian sentence is replaced with a less ambiguous equivalent in a discontinuous context, Systran translates this part of the sentence correctly:

(2) Законодательная власть **вотирует** сотни законов ежегодно.
(Literal translation: Legislative power **votes** hundreds of **laws** yearly)
→Legislative authority votes hundreds of laws yearly.

The Russian input in (2) is much less idiomatic than in (1), but native English speakers find the MT output much easier to understand.

5.2 Evaluation-based requirements for MT-tractable language

Traditionally Controlled Language for MT and MT-tractable language are viewed as universal concepts, which should ideally work for any MT system. Specifications for such language have hitherto been derived from general considerations about MT, language complexity and ambiguity, or from results obtained on test suites (Bernth and Gdaniec, 2001: 177-195), but not from a corpus-level evaluation of particular MT engines. However, it is reasonable to expect that many such requirements would not stand the test of this type of evaluation. On the other hand, new requirements not envisaged by intuitive considerations may be discovered via corpus-based MT evaluation.

Our experiment supports the argument for the development of evaluation-guided methods for deriving specifications for tractable language, since blind rewriting caused deterioration for 37% of rewritten constructions. This result implies that there is no universal concept of MT-tractability, but that 'tractability' depends on the performance of particular MT engines. The proportion of constructions difficult for all MT systems is much smaller than expected and it is hard to justify any system-independent requirements for MT-tractable language. A challenge is to derive such specifications and rewriting rules automatically, based on evaluation of particular MT systems.

5.3 Construction-oriented MT evaluation

In this context, automated evaluation methods should not only give a general indication of MT quality, but also identify poorly-translated constructions, so BLEU-type scores alone will not be sufficient. In fact, the improvements analysed here may have a negligible effect in terms of BLEU scores, but still they can have an impact on the comprehensibility of many frequent translation contexts, so the proposed methodology can help to some extent to automate the process of error analysis for frequently mistranslated linguistic constructions.

In this paper we have demonstrated the use of concordance-based evaluation, which can be also modified for use in an automated evaluation framework, as suggested in (Anon, 2008), and used for assessing translation quality of particular constructions. As a next stage, synonymous constructions which are more tractable for a given MT system can be found automatically using distributional similarity techniques for multiword expressions, such as those proposed in (Anon, 2007). These constructions then become candidates for automatic rewriting.

6 Conclusions

Rewriting of LVCs can greatly improve the comprehensibility of their translations. In our experiment we achieved a reduction in incomprehensible translations of around 40%. The experiment also suggests that there is no universal concept of MT-tractability, so rewriting of contexts for problematic constructions should be guided by evaluation of the performance of particular MT systems for those constructions.

Future work will involve developing an automated approach to identifying ambiguous lexical units and problematic constructions in MT and finding their MT-tractable counterparts with similar distribution.

Automatic rewriting can be developed as a pre-processing functionality for users of state-of-the-art MT systems, and also as stand-alone rewriting applications, e.g., for pivot MT architecture via closely-related languages (Babych et al, 2007), where MT-tractable language can be viewed as closely related to the source.

References

Babych, Bogdan, Anthony Hartley, and Serge Sharoff. 2007. Translating from under-resourced languages: comparing direct transfer against pivot translation. In *Proceedings of the MT Summit XI*, pages 412–418, Copenhagen.

Bernth, Arendse and Claudia Gdaniec. 2001. *MTranslatability. Machine Translation*, 16:175–218.

Mel'čuk, Igor A. 1996. Lexical Functions: a tool for the description of lexical relations in a lexicon. In Leo Wanner, editor, *Lexical Functions in Lexicography and Natural Language Processing*, pages 37–102. John Benjamins.

Mudraya, Olga, Scott S. L. Piao, Paul Rayson, Serge Sharoff, Bogdan Babych, and Laura Lofberg. 2008. Automatic extraction of translation equivalents of phrasal and light verbs in English and Russian. In S. Granger and F. Meunier, editors, *Phraseology: an interdisciplinary perspective*, pages 293–309. John Benjamins.

Nyberg, Eric, Teruko Mitamura, and Willem-Olaf Huijsen. 2003. Controlled language for authoring and translation. In Harold Somers, editor, *Computers and Translation. A translator's guide*, pages 245–281. John Benjamins.

Salkoff, Morris. 1999. *A French-English Grammar: a contrastive grammar on translational principles*. John Benjamins.

Somers, Harold. 1997. A practical approach to using machine translation software. *The Translator*, 3(2):193–212.

Ye, Yang, Ming Zhou, and Chin-Yew Lin. 2007. Sentence Level Machine Translation Evaluation as a Ranking Problem: one step aside from BLEU. In: *Proceedings of the Second Workshop on Statistical Machine Translation,* Prague.

Learning Labelled Dependencies in Machine Translation Evaluation

Yifan He and **Andy Way**
Centre for Next Generation Localisation
School of Computing
Dublin City University
{yhe,away}@computing.dcu.ie

Abstract

Recently novel MT evaluation metrics have been presented which go beyond pure string matching, and which correlate better than other existing metrics with human judgements. Other research in this area has presented machine learning methods which learn directly from human judgements. In this paper, we present a novel combination of dependency- and machine learning-based approaches to automatic MT evaluation, and demonstrate greater correlations with human judgement than the existing state-of-the-art methods. In addition, we examine the extent to which our novel method can be generalised across different tasks and domains.

1 Introduction

There is no doubt that the onset of automatic evaluation metrics such as BLEU (Papineni et al., 2002) has led directly to improvements in quality in machine translation (MT). Prior to their introduction, most results were anecdotal, or researchers had to conduct expensive human evaluations in order to validate their work.

However, seven years after their introduction, there is widespread recognition in MT that these string-based metrics are not discriminative enough to reflect the translation quality of today's systems, many of which have gone beyond *n*-grams (cf. (Callison-Burch et al., 2006)).

With that in mind, a number of researchers have come up with metrics which are not wholly string-based. Perhaps the best-known alternative metric is METEOR (Banerjee and Lavie, 2005), which while still being string-based, tries to improve on the matching schemes of BLEU by incorporating synonym matching via WordNet.

Given that many of today's MT systems incorporate some kind of syntactic information (e.g. (Chiang, 2005)), it was perhaps natural that other researchers would seek to use syntax in automatic MT evaluation as well. The first step in this direction was by (Liu and Gildea, 2005), who used syntactic structure and dependency information in order to see past the surface phenomena. Two of these metrics are based on matching syntactic subtrees between the translation and the reference, and the third is based on matching headword chains, but only for *unlabelled* dependencies. Since then, (Owczarzak et al., 2007a; Owczarzak et al., 2007b) have extended this line of research with the use of a term-based encoding of LFG *labelled* dependency graphs into unordered sets of dependency triples, and calculating precision, recall, and f-measure on the sets corresponding to the translation and reference sentences. With the addition of partial matching and *n*-best parses, (Owczarzak et al., 2007a; Owczarzak et al., 2007b) considerably outperform Liu and Gildea's (2005) highest correlations with human judgement.

Another line of research has led to machine learning methods which learn directly from human judgements (Ye et al., 2007). In this paper, we combine the syntax (dependency)-based and the machine learning-based approaches, and show greater correlations with human judgement than (Owczarzak et al., 2007a; Owczarzak et al., 2007b). We use both Ranking and Regression Support Vector Machines (SVMs) (Burges, 1998) in a range of experiments on different language pairs and data sets. We also examine the extent to which our novel method can be generalised across differ-

Proceedings of the 13th Annual Conference of the EAMT, pages 44–51,
Barcelona, May 2009

ent tasks and domains.

The remainder of the paper is organised as follows. In section 2, we outline approaches to automatic MT evaluation which are relevant to our work. In particular, in section 3 we describe the LFG labelled dependency approach of (Owczarzak et al., 2007a; Owczarzak et al., 2007b). In section 4, we demonstrate how labelled dependencies can be matched using SVMs, and describe the range of experiments carried out in section 5. The paper ends with our concluding remarks together with avenues for further research.

2 Evaluation Metrics in MT

Automatic evaluation metrics enable researchers to validate and optimise translation methods quickly. Simple *n*-gram-based metrics such as BLEU (Papineni et al., 2002) are fundamental to the development and tuning of MT systems. However, these *n*-gram-based metrics suffer from several shortcomings, such as low correlation with human judgement on the sentence level, exhibiting a bias towards statistical systems (Callison-Burch et al., 2006), and inconsistency in related evaluation scenarios (Chiang et al., 2008).

Many approaches have been taken to overcome the insufficiencies of BLEU. Word-based metrics like METEOR (Banerjee and Lavie, 2005) try to improve on the matching scheme; paraphrase-based methods such as ParaEval incorporate paraphrases extracted from an external data source (Zhou et al., 2006); syntactic methods try to use syntax information in hypothesis and reference (cf. section 2.1); and machine learning methods learn directly from human judgements (cf. section 2.2).

2.1 Dependency-based Metrics

The shortcomings of *n*-gram metrics have led a number of researchers to exploit more grammatical information in the hypothesis and reference sentences.

Syntactic features were first introduced in MT evaluation in (Liu and Gildea, 2005), who developed several metrics using constituency or dependency structure. (Owczarzak et al., 2007a; Owczarzak et al., 2007b) improved on the dependency matching of (Liu and Gildea, 2005) by using *n*-best labelled dependency triples produced by an LFG parser, so that parser noise is reduced and partial matchings can be found. (Kahn et al., 2008) match *n*-best head-modifier dependencies extracted from *n*-best constituency parses. They also consider the probabilities given by the constituency parser.

Dependency information is also used in metrics that incorporate different information sources. (Giménez and Màrquez, 2008) experimented using different levels of linguistic features and dependency relation-based metrics are among their best metrics at both system and sentence levels. Machine learning metrics such as (Ye et al., 2007) and (Albrecht and Hwa, 2007) also use some head-modifier dependency matches or dependency chains as features.

2.2 Machine Learning-based Metrics

Three kinds of machine learning-based approaches have been used in MT evaluation: (i) *Classification*-based approaches (Corston-Oliver et al., 2001) train a classifier to discriminate between the reference and the hypothesis. The higher the likelihood of a hypothesis being a reference, the better its quality is assumed to be; (ii) *Regression*-based methods (Albrecht and Hwa, 2007) train a model to try to reproduce the human judgement scores for each translation hypothesis; (iii) *Ranking*-based approaches (Ye et al., 2007) train a model with the ranking of different hypotheses on a particular sentence instead of the values of the scores.

Among these three approaches, classification only captures the difference between the hypotheses and the reference but ignores any differences in quality among these hypotheses. Both ranking- and regression-based methods have been reported to be successful in various MT evaluation tasks. In our experiments we combine them with the dependency-based method of (Owczarzak et al., 2007a; Owczarzak et al., 2007b) and directly compare them in a ranking task.

3 LFG Labelled Dependencies

Our work extends the method of (Owczarzak et al., 2007a; Owczarzak et al., 2007b) who use labelled dependencies in Lexical-Function Grammar (LFG). In LFG, a sentence is represented in both a hierarchical tree structure (C-structure) which captures the organisation of a sentence, and a set of labelled dependencies (F-structure). The dependencies in LFG are attribute-value features such as `subj(arrive, Julie)` or `pers(Julie,`

3) which capture the grammatical relations between constituents. They are more precise than head-modifier unlabelled dependencies. Here the trigram (DEP, HEAD, MODIFIER) is called a triple. In subj(arrive, Julie), DEP is subj, HEAD is arrive and MODIFIER is Julie.

In (Owczarzak et al., 2007a; Owczarzak et al., 2007b) it is shown that LFG F-structures can capture variations between sentences. For example, "Julie arrived yesterday." and "Yesterday Julie arrived." have only one bigram (Julie, arrived) in common but the same F-structures. This feature can help us better judge how similar a reference sentence and a hypothesis sentence are in MT evaluation.

3.1 Matching of Dependency Triples

To utilise LFG dependencies in MT evaluation, we use the LFG parser described in (Cahill et al., 2004) to generate dependency triples and perform matching on the triples. A hypothesis sentence is considered of higher quality when it has more triples matched with the reference sentence.

We perform three kinds of dependency matchings in our experiment: exact matching, partial matching, and WordNet extended matching. In exact matching all three elements in the triple must be the same to complete a match. With respect to the previous example, in partial matching, two triples can have different HEAD or MODIFIER values, whereas in WordNet extended matching, HEAD and MODIFIER can be substituted by synonyms in WordNet.

We only perform partial and WordNet extended matching on PREDICATE-ONLY dependencies (Owczarzak et al., 2007a; Owczarzak et al., 2007b). Both exact and partial matches on dependency type x are counted as one match on type x. A WordNet extended match is counted as one match on type x_WN.

3.2 Parser Noise and Matching in *n*-best Parses

The outputs of MT systems are often syntactically ill-formed and this makes it difficult for parsers to generate plausible parses. To compensate for this problem, we parse the hypothesis and reference translations to obtain the 50-best parses of each. Using the 50-best parses increases the chance of finding the correct match between the hypotheses and references.

For each pair of parses, we match the dependency triples, and select the pair of parses that has the highest F-score (cf. (3)) as matching and output the matching detail of this pair. Details on the effect of multiple parses can be found in (Owczarzak et al., 2007a; Owczarzak et al., 2007b).

3.3 Calculation of Matching Percentage

There are two ways of normalising the number of matchings. We can normalise with respect to the total number of triples in the hypothesis sentence (precision matching), as in (1):

$$P = \frac{\#matching\_triples}{\#triples\_in\_hypothesis} \quad (1)$$

or the total number of triples in the reference sentence (recall matching), as in (2):

$$R = \frac{\#matching\_triples}{\#triples\_in\_reference} \quad (2)$$

In (Owczarzak et al., 2007a; Owczarzak et al., 2007b), precision matching and recall matching are combined into an F-score, as in (3):

$$F = \frac{2PR}{P+R} \quad (3)$$

When using this combination, the relative weights of precision and recall are implicitly set to 1:1. In our experiment this combination is not necessary, as we can use both precision and recall values as features and let the SVM determine the respective weights of precision and recall.

4 Combining Labelled Dependency Matches with SVM

4.1 SVM in MT Evaluation

We use Ranking and Regression Support Vector Machines (Burges, 1998) in our experiments. Both Ranking and Regression SVMs assign a score to an input instance z, as in (4):

$$f(z) = \sum_{i=1}^{m} \alpha_i y_i \Phi(x_i) \cdot \Phi(z) + b \quad (4)$$

where (x_i, y_i) is the training example and Φ is the transformation function which transforms the input space to the feature space. However, the quantitative value from a ranking SVM is meaningless and only indicates its ranking.

The output of a ranking SVM aims at producing the correct rank of input examples, whereas regression SVMs aim at producing a value corresponding

to the input. Thus the ranking SVM maximises τ on a training set, where r_i is the metric ranking of systems on sentence i and r_i^* is the human ranking on sentence i, as in (5):

$$\frac{1}{n}\sum_{i=1}^{n}\tau(r_i, r_i^*) \quad (5)$$

Note that Kendall's τ measures the relevance of two rankings: $\tau(r_a, r_b) = \frac{P-Q}{P+Q}$, where P and Q are the amount of concordant and discordant pairs in r_a and r_b.

Regression SVMs, by contrast, are directly modelled on the human judgement scores by minimising (6):

$$\frac{1}{2}\sum_{i=1}^{m}(y_i - f(x_i))^2 \quad (6)$$

4.2 Kernels of SVM

We can often find a kernel function K in (4) with $K(x, z) = \Phi(x) \cdot \Phi(z)$. Kernel functions implicitly transform the input space into the feature space, while computation is still done in the input space.

We use three kinds of kernels in our experiments: (i) *Linear* kernels, the simplest form of kernel which do not transform the input space:

$$K(x, z) = x \cdot z \quad (7)$$

(ii) *Polynomial* kernels:

$$K(x, z) = (\alpha + \beta x \cdot z)^p \quad (8)$$

In our experiments α and β are set to 1, and p is set to 3.

(iii) *Radial Basis Function (RBF)* kernels:

$$K(x, z) = exp(-\gamma||x - z||^2) \quad (9)$$

The RBF kernel is the most complex kernel of the three. In some NLP tasks such as text categorisation (Joachims, 1998), RBF kernels are shown to capture the characteristics of the training data more accurately than linear or polynomial kernels. In our experiments γ is set to 1.

4.3 Normalisation of Features

The features in our experiments are the matching percentages on different types of dependencies. We propose two ways of normalising the value: horizontal and vertical. In horizontal normalisation, the number of matches on a certain dependency type are normalised by the total number of triples in the test/reference (based on whether precision or recall dependency matching is used) sentence, as in (10):

$$H(i) = \frac{\#matching\_depType(i)}{\#allTypes} \quad (10)$$

In vertical normalisation, only the number of dependencies of the same type are considered, as in (11):

$$V(i) = \frac{\#matching\_depType(i)}{\#depType(i)} \quad (11)$$

In horizontal normalisation, dependency types x and x_WN are counted separately. However, in vertical normalisation x_WN is counted as x, as x_WN is produced during matching, and we do not have this dependency type in the test or reference sentences.

Our horizontal normalisation is equivalent to the approach of (Ye et al., 2007). The vertical normalisation is a more radical approach to reflect the relative ratio of matches on different dependency types.

5 Experiments

5.1 Data

We use two data sets in our experiments. We use the WMT08 evaluation shared task dataset for Ranking SVM training and testing. We use 3,249 human rankings on outputs from different MT systems. The rankings are just a reflection of the relative quality of these systems; no absolute scores are given. We use 177 sentences from the Czech–English News Commentary task and 123 sentences from the Czech–English News task as our development set (DEV). We use 358 Czech–English News task sentences as the test set (TEST).

For the regression SVM we use the MTC4 corpus from LDC. The corpus consists of human-assigned fluency and adequacy scores to 11,028 outputs of MT systems. We remove the outputs that cause parser errors, leaving 11,004 segments, of which 2,000 sentences are used as our DEV set, 2,004 are used as the TEST set and the remaining 7,000 are used for training.

For generalisablity testing we also run experiments on WMT08 data with regression models generated from MTC4 data, and we run cross-language and cross-domain tests on WMT08 data.

Table 1: Ranking SVM: Different Kernels. Cons.: Consistency percentage; Corr.: Spearman's coefficient

	Cons. DEV	Corr. DEV	Cons. TEST	Corr. TEST
BLEU-4	0.3397	0.1896	0.2515	0.1297
BLEU-4s	0.5251	0.0909	0.5480	0.1427
LFG-F	0.5892	0.2521	0.5565	0.1796
PR-HV-L	**0.6325**	**0.2753**	**0.6055**	**0.2057**
PR-HV-P	0.6083	0.2548	0.5202	0.0806
PR-HV-R	0.5667	0.2008	0.5117	-0.0006

Table 2: Ranking SVM: Different Data Representation

	Cons. DEV	Corr. DEV	Cons. TEST	Corr. TEST
BLEU-4	0.3397	0.1896	0.2515	0.1297
BLEU-4s	0.5251	0.0909	0.5480	0.1427
LFG-F	0.5892	0.2521	0.5565	0.1796
P-V-L	0.5632	0.1599	0.5373	0.1227
P-H-L	0.5407	0.1315	0.5437	0.1565
R-V-L	0.6152	**0.2815**	0.5287	0.1656
R-H-L	0.5771	0.1988	0.5309	0.1903
PR-V-L	0.6170	0.2686	0.5884	**0.2206**
PR-H-L	0.6048	0.2178	**0.6055**	0.1939
P-HV-L	0.5685	0.1764	0.5415	0.1039
R-HV-L	0.6153	0.2751	0.5522	0.2068
PR-HV-L	**0.6325**	0.2753	**0.6055**	0.2057

5.2 Experimental Settings

We tested the ranking SVM with different types of feature representation. Normalisation (Norm) is performed with the horizontal (H), vertical (V) or both (HV) methods. Dependency matching (DEP) is computed in terms of precision (P), recall (R) or both (PR). We test with SVMs of linear (L), polynomial (P) and RBF (R) kernels (KERNEL) using the SVMLight software. Each configuration is denoted with {NORM}-{DEP}-{KERNEL} in both ranking and regression experimental results.

We use the following three metrics as baselines: BLEU (BLEU-4), add-one BLEU (BLEU-4s) and the labelled LFG-based metric (LFG-F) as described in (Owczarzak et al., 2007a; Owczarzak et al., 2007b). Note that the result of the LFG-F metric would have among the highest correlations with human judgement in the WMT08 shared evaluation task.

5.3 Experiments on Ranking SVM

We train ranking SVMs on WMT08 data to produce rankings of different system outputs on the same sentence.

Usually, the correlation between a metric and human rankings can be measured by Spearman's rank order correlation, defined in (12), where d is the difference between corresponding values in rankings and n is the length of the rankings:

$$\rho = 1 - (\frac{6 \sum d^2}{n(n^2-1)}) \quad (12)$$

However, in (Callison-Burch et al., 2008), it is argued that averaging ρ is meaningless, and so pair-wise consistent percentage is used instead to measure correlations in the WMT08 shared evaluation task. The pair-wise consistent percentage is equal to the number of correct pair-wise comparisons made by a metric divided by the total number of pair-wise comparisons to make.

We report both consistent percentage and sentence-level Spearman's correlation in our experiments. The Spearman's correlation is first computed on each ranking, and then averaged.

We explore the choice of different {KERNEL}s (Table 1) with PR-HV data representation (the best representation) and the choice of different {NORM}alization and {DEP}endency matching schemes (Table 2) with linear kernel (the best kernel).

In our experiments, PR-HV-L, the metric that uses all variations of features, yields the best overall results and outperforms the baseline on both DEV and TEST sets. A number of observations present themselves: (i) In Table 2, recall-based dependency match rates appear to be better features than precision-based rates. This pattern is also observed in other metrics such as METEOR. This is another example of the importance of recall in MT evaluation; (ii) In Table 1, more sophisticated kernels such as Polynomial and RBF kernels do not increase the performance of the metric and sometimes even decrease it. This might appear surprising, yet recall that we reserved all Czech–English translations for the development and test sets, so the SVM is not exposed to any human judgements on this language pair during training. We did this in order to show the generality of our machine learning-based method, but in so doing we may have caused the more sophisticated kernels to overfit on other language pairs. It tells us that selection of features is more important for our method than the learning algorithm itself; (iii) Vertical match features produce some good results but are more prone to overfitting. Using the RBF kernel on vertical match features often leads to lower correlations. The problem with the vertical match feature is that it ignores the total number of dependencies

Table 3: Regression SVM: Different Kernels. F/A Corr.: Correlation on fluency/adequacy

	F Corr. DEV	A Corr. DEV	F Corr. TEST	A Corr. TEST
BLEU-4	0.0679	0.145	0.1179	0.2087
BLEU-4s	0.0919	0.2077	0.1724	0.2499
LFG-F	**0.1076**	0.2926	0.2453	0.3779
R-H-L	0.0812	0.2987	**0.2506**	**0.3992**
R-H-P	0.0869	**0.2998**	0.2322	0.3948
R-H-R	0.0880	0.2996	0.2302	0.3935

Table 4: Regression SVM: Different Data Representation

	F Corr. DEV	A Corr. DEV	F Corr. TEST	A Corr. TEST
BLEU-4	0.0679	0.145	0.1179	0.2087
BLEU-4s	0.0919	0.2077	0.1724	0.2499
LFG-F	**0.1076**	0.2926	0.2453	0.3779
P-V-L	0.0961	0.2025	0.1993	0.2723
P-H-L	0.1030	0.2331	0.2040	0.2723
R-V-L	0.0694	0.2698	0.2222	0.3894
R-H-L	0.0812	0.2987	**0.2506**	0.3992
PR-V-L	0.0793	0.2669	0.2189	0.3827
PR-H-L	0.0989	**0.3027**	0.2436	0.3934
P-HV-L	0.1040	0.2165	0.2112	0.2894
R-HV-L	0.0850	0.2867	0.2307	**0.3999**
PR-HV-L	0.0933	0.2828	0.2288	0.3911

in a sentence. As a result, an output that correctly translates `subj` in a simple sentence with 2 dependencies will receive the same score as an output that only translates `subj` correctly in a compound sentence of 20 dependencies. This leads to problematic features and the problem might be exacerbated during learning; and (iv) When H, V, P and R are all used as features, we obtain the best overall result. This suggests that our different methods of normalisation and dependency matching are complementary in our ranking experiment.

5.4 Experiments on Regression SVM

In the regression SVM experiment, we use SVM to learn the scores which are assigned by human judges. The models for predicting fluency and adequacy scores are trained separately.

We calculate Pearson's correlation on both fluency and adequacy. Pearson's correlation is defined as:

$$r = \frac{1}{n-1} \sum (\frac{x_i - \bar{X}}{s_X})(\frac{y_i - \bar{Y}}{s_Y}) \qquad (13)$$

where x_i is the value of the i^{th} score, $\bar{X}$ is the mean score and s_X is the standard deviation.

The results on different kernels and different data representation are reported in Table 3 and Table 4 respectively. For the regression task, we test the choice of kernels on the R-V representation, which performs better than PR-HV in this task. In this experiment, we do not see a particular metric that consistently outperforms the baseline with respect to fluency. However, all metrics that are based on horizontal normalisation and recall-style dependency matching perform better than the baseline with respect to adequacy, for several reasons. Firstly, the features of our SVM models are the decomposed parts of LFG-F. LFG-F is better at evaluating adequacy than fluency (Owczarzak et al., 2007a; Owczarzak et al., 2007b). Thus we have better features for our adequacy-predicting SVM model.

Secondly, note that the fluency correlation on the DEV set is generally at a very low level, which indicates that the sentences in our DEV set are very hard to judge with respect to fluency. At this level, many trivial reasons can lead to an increase or decrease in correlation. In general, we can consider our R-H-L and PR-H-L metrics to be on a par with the baseline as far as fluency is concerned.

Except for the variance in fluency and adequacy, many tendencies observed in our ranking experiment still apply here. The recall-based features still prevail and sophisticated kernels do not improve performance. Vertical normalisation has a bigger negative impact in this experiment. It suggests that regression is more error-prone than ranking, perhaps because regression is harder.

5.5 Cross-Task Generalisability

We choose the two best-performing (R-H-L, PR-H-L) as well as two somewhat mediocre (R-HV-L, PR-HV-L) regression models and use them to compute scores for our ranking DEV and TEST set. We do not run this experiment in the opposite direction, because the MTC4 data is not collected in a ranking scenario and we consider it incomparable to the results on WMT08. We calculate Spearman's coefficient between the rankings induced from these regression scores and the human rankings to validate the generalisability of our learning method. For regression SVMs trained on MTC4, WMT08 is a corpus that is different with respect to language pair, domain, and evaluation criterion. The results are shown in Table 5.

Basically all four metrics trained on MTC4 outperform the LFG F-Score baseline on the TEST set, but are on a par or inferior on the DEV set. We consider this tendency to be related to the differ-

Table 5: Cross-Task Experiments

	Cons. DEV	Corr. DEV	Cons. TEST	Corr. TEST
BLEU-4	0.3397	0.1896	0.2515	0.1297
BLEU-4s	0.5251	0.0909	0.5480	0.1427
LFG-F	**0.5892**	**0.2521**	0.5565	0.1796
R-H	0.5875	0.2269	0.5714	**0.2471**
PR-H	0.5823	0.2152	**0.5991**	0.2084
R-HV	0.5649	0.1526	0.5479	0.1931
PR-HV	0.5719	0.1700	0.5714	0.1638

Table 6: Cross-Language Pair Experiments

	French		Other	
	Cons.	Corr.	Cons.	Corr.
BLEU	0.2795	0.1913	0.3652	0.1827
BLEU-4s	0.5818	0.2255	0.5675	0.1980
LFG-F	0.6204	0.2550	**0.5994**	**0.2503**
PR-V-L	0.6159	0.2420	0.5813	0.1848
PR-H-L	**0.6522**	**0.3131**	0.5844	0.2118
PR-HV-L	0.6227	0.2706	0.5896	0.1931

ence in domains. The ranking DEV set is dominated by commentary data, but the TEST set consists of news data only, which is identical to the MTC4 corpus we use to train the Regression SVM.

The results show that our method is generalisable to different tasks and evaluation criteria. When tested on similar domains, our regression SVM not only performs better than a very high baseline, but also approaches the performance of the best SVM trained specially for Ranking. Furthermore, the better performing metrics on MTC4 continue to perform well on WMT08.

However, our method is quite sensitive to domain change. The regression SVM trained on a completely different domain performs worse than the Ranking SVM on the DEV set, whereas on the TEST set it performs better than the Ranking SVM, which is trained on a multi-domain corpus.

5.6 Cross-Language Pair and Cross-Domain Generalisability

We carried out more experiments on the WMT08 data to explore the generalisability of our method over different language pairs and different domains. As far as language pair generalisability is concerned, we divide the dataset by language pairs into French–English and Other–English parts. We train the metrics on half of the French–English data, and test the model on the other half as well as Other–English data. The results are provided in Table 6.

For domain generalisability, we train the metrics on half of the News data and test them on the other half, as well as non-News data. The results are shown in Table 7. In both experiments we test with three metrics: PR-V-L, PR-H-L and PR-HV-L.

Table 7: Cross-Domain Experiments

	News		Non-News	
	Cons.	Corr.	Cons.	Corr.
BLEU	0.3035	0.1653	0.4739	0.2906
BLEU-4s	0.5548	0.2013	0.6277	0.2992
LFG-F	0.6112	0.2905	**0.6313**	**0.3007**
PR-V-L	0.6102	0.2540	0.5858	0.2088
PR-H-L	**0.6208**	**0.2957**	0.6129	0.2745
PR-HV-L	0.6134	0.2694	0.5996	0.2285

In both tests our methods do not outperform the baseline on different language pairs or domains. This is because our training set is very small. We are actually using a model trained on just hundreds of samples to rank thousands of samples in a different language pair/domain. In this context, all the tested methods obtain consistent percentages very close to the baseline in the cross-language pair experiment. It confirms that our method is more generalisable over different language pairs, and is somewhat more sensitive to changes in domains.

The shortcomings of vertical normalisation are magnified in these experiments. The correlations of our metrics on out-of-domain test sets follows the pattern of H $>$ HV $>$ V, which indicates that vertical normalisation causes performance to deteriorate. It accords with our assumption in the regression experiment that vertical normalisation is more prone to error on harder tasks.

6 Conclusion and Further Work

In this paper, we have presented a novel approach to automatic MT evaluation, where the labelled dependency approach of (Owczarzak et al., 2007a; Owczarzak et al., 2007b) is combined with the use of both Ranking and Regression Support Vector Machines (SVMs) (Burges, 1998). In our approach, we learn the required labelled dependencies, and show that our method improves over the approach of (Owczarzak et al., 2007a; Owczarzak et al., 2007b) with respect to correlation with human judgements. In addition, we demonstrate that our method is generalisable over different language pairs, but is somewhat more sensitive to changes in domains.

As far as extensions to this work are concerned, we aim to experiment with more features to improve cross-domain adaptability and to prevent any

overfitting. In addition, a more in-depth analysis needs to be carried out in order to discover which particular features contribute most to the correlation with human judgement.

Acknowledgements

We are grateful to Science Foundation Ireland (`http://www.sfi.ie`) grant number 07/CE/I1142 for sponsoring this research.

References

Albrecht, Joshua and Rebecca Hwa. 2007. Regression for sentence-level mt evaluation with pseudo references. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 296–303, Prague, Czech Republic.

Banerjee, Satanjeev 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, MI.

Burges, Christopher J. C. 1998. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167.

Cahill, Aoife, Michael Burke, Ruth O'Donovan, Josef Van Genabith, and Andy Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 319–326, Barcelona, Spain.

Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of BLEU in machine translation research. In *EACL-2006, 11th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings*, pages 249–256, Trento, Italy.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, OH.

Chiang, David, Steve DeNeefe, Yee Seng Chan, and Hwee Tou Ng. 2008. Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 610–619, Honolulu, HI.

Chiang, David. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, MI.

Corston-Oliver, Simon, Michael Gamon, and Chris Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proceedings of 39th Annual Meeting and 10th Meeting of the European Chapter of the Association for Computational Linguistics*, pages 148–155, Toulouse, France.

Giménez, Jesús and Lluís Màrquez. 2008. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 195–198, Columbus, OH.

Joachims, Thorsten. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, pages 137 – 142, Berlin/Heidelberg. Springer.

Kahn, Jeremy G., Mari Ostendorf, and Brian Roark. 2008. Automatic syntactic MT evaluation with expected dependency pair match. In *Proceedings of the Workshop Metrics MATR - Metrics for Machine Translation Challenge, Eighth Conference of the Association for Machine Translation in the Americas*, Waikiki, HI.

Liu, Ding and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, MI.

Owczarzak, Karolina, Josef van Genabith, and Andy Way. 2007a. Evaluating Machine Translation with LFG Dependencies. *Machine Translation*, 21(2):95–119.

Owczarzak, Karolina, Josef van Genabith, and Andy Way. 2007b. Labelled dependencies in machine translation evaluation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 104–111, Prague, Czech Republic.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Ye, Yang, Ming Zhou, and Chin-Yew Lin. 2007. Sentence level machine translation evaluation as a ranking. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 240–247, Prague, Czech Republic.

Zhou, Liang, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. Paraeval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 447–454, New York City, NY.

Improving a Catalan-Spanish Statistical Translation System using Morphosyntactic Knowledge

Mireia Farrús, Marta R. Costa-jussà, Marc Poch, Adolfo Hernández, and José B. Mariño
TALP Research Centre, Department of Signal Theory and Communications
Universitat Politècnica de Catalunya
C/ Jordi Girona 1-3, 08034 Barcelona, Spain
{mfarrus,mruiz,mpoch,adolfohh,canton}@gps.tsc.upc.edu

Abstract

In this paper, a human evaluation of a Catalan-Spanish Ngram-based statistical machine translation system is used to develop specific techniques based on the use of grammatical categories, lexical categorisation and text processing, for the enhancement of the final translation. The system is successfully improved when testing with ad hoc and general corpora, as it is shown in the final automatic evaluation.

1 Introduction

Statistical Machine Translation (SMT) nowadays has become one of the most popular Machine Translation paradigms. The SMT approach allows to build a translator with open-source tools as long as a parallel corpus is available. If the languages involved in the translation belong to the same linguistic family, the translation quality can be surprisingly nice. Furthermore, one of the most attractive reasons to build an statistical system instead of an standard rule-based system is the little human effort required.

Theoretically, when using SMT, no linguistic knowledge is required. In practice, once the system is built and specially, if the translation quality is high, then the linguistic knowledge becomes necessary to make further improvements (Niessen and Ney, 2000; Popović and Ney, 2004; Popović et al., 2006). In fact, the main question that arose at the beginning of this work was: which are the steps to follow when the intention is to improve a high quality statistical translation?

Let's consider a high quality statistical translation defined as the system which has a BLEU around 75% with a single reference in an in-domain test. This is a relatively unusual situation as most of the statistical translation systems have much lower performance. This study is devoted to develop this stage in the Catalan-Spanish pair in both directions.

The study starts from a high quality Ngram-based statistical translation baseline system, trained with the aligned Spanish-Catalan parallel corpus taken from *El Periódico* newspaper, which contains 1.7 million sentences. A human error analysis of the translation is then performed and used to further improve the translation by introducing statistical techniques and linguistic rules.

This paper is organised as follows. Section 2 describes the Ngram-based statistical translation system used as baseline system. Section 3 reports the human error analysis and evaluation of the baseline system, whose solutions based on statistical techniques, linguistic rules and text processing are explained in section 4. In section 5, an automatic evaluation of the new system is performed and discussed. Finally, Section 6 sums up the conclusions.

2 Ngram-based statistical translation system

An Ngram-based SMT system regards translation as a stochastic process. In recent systems, such an approach is faced using a general maximum entropy approach in which a log-linear combination of multiple feature functions is implemented (Och, 2003). This approach leads to maximising a linear combination of feature functions:

$$\tilde{t} = \underset{t}{argmax} \left\{ \sum_{m=1}^{M} \lambda_m h_m(t,s) \right\} \qquad (1)$$

where the *argmax* operation denotes the search

Proceedings of the 13th Annual Conference of the EAMT, pages 52–57,
Barcelona, May 2009

problem, i.e. the generation of the output sentence in the target language, $h_m(t, s)$ are the feature functions and λ_m are their corresponding weights.

The main feature function (and the only one in our baseline system) is the Ngram-based translation model which is trained on bilingual n-grams. This model constitutes a language model of a particular bi-language composed of bilingual units (translation units) which are referred to as tuples. In this way, the translation model probabilities at the sentence level are approximated by using n-grams of tuples.

The Ngram-based approach is monotonic in that its model is based on the sequential order of tuples during training. Therefore, the baseline system with only one feature function may be specially appropriate for pairs of languages with relatively similar word order schemes. Further details can be found in Mariño (2006 et al.).

3 Linguistic error analysis

In this section we report the linguistic error analysis performed over the Ngram-based baseline output. The analysis was performed by a Catalan and Spanish native linguist at the level of syntaxis, semantics and morphology using out-of-domain text. The set of errors are listed and briefly described next.

Obligation The obligation Spanish expression *tener que* (*have to*) was literally translated as **tenir que* into Catalan, instead of *haver de*.

***Solo* confusion** The term *solo* in Spanish can be related to three distinct parts of speech (POS): adverb (*only*), adjective (*alone*) or noun (*solo*). Since the translation into Catalan depends on the POS, the translated term becomes erroneous when the Spanish POS is not well-recognised, which happens specially in this case between the adverb and the adjective.

Apostrophe In the Spanish-Catalan translation, the apostrophe rules for the Catalan articles *el*, *la* and the preposition *de* in front of vowels are not fulfilled.

Geminated *l* (*l·l*) Although the Catalan geminated *l* should be always written with a middle dot (·), it is very frequent to find it written with normal dot, which leads to erroneous translations into Spanish.

Omission of prepositions The preposition *de* is frequently omitted when translating the Spanish verb *deber* (*must*) into the phrasal verb *haver de*. On the other hand, Spanish normally uses the preposition *a* in front of a direct object while Catalan does not, so that such preposition is usually omitted in the Catalan-Spanish translation.

***a*, *en* prepositions** These prepositions are used in very distinct ways in both Catalan and Spanish languages, so that it becomes difficult to achieve correct translations in both directions.

Possessive pronouns and adjectives In Catalan, possessive pronouns and adjectives are expressed with the same term, whereas Spanish does not. This ambiguity in Catalan leads to confusion in the translation to Spanish.

Conjunction *perquè* This conjunction is ambiguous in the Catalan-Spanish translation since it depends on whether the conjunction is causal, in which case corresponds to *porque* (because), or final, where it corresponds to *para que* (in order to).

Verb *soler* The conjugated forms *sol* and *sols* of the verb *soler* (to use to) can be confused by the adjective meaning *alone* that uses the same term.

Conjunctions *i,o* These Catalan conjunctions must be translated into Spanish as *e* and o instead of *y* and *u* when the following word begins with *i* and *u*, respectively.

Numbers Many numeric expressions are not included in the training corpus, so that no translation can be generated in any of the target languages.

Hours Catalan and Spanish time expressions differ significantly, being usually impossible to use literal translations. The main difference is found in the use of the quarters: where Spanish hours express the quarters that pass from a specific hour, Catalan uses the following hour. E.g. *Las cuatro y cuarto* (four and a quarter) in Spanish would correspond to *Un quart de cinc* (a quarter of five) in Catalan.

Pronominal clitics Frequently, the translation fails in the combination of the pronominal clitic and the corresponding verb.

***Cuyo* relative pronoun** The relative constructions involving the Spanish pronoun *cuyo* are subject to a lexical reordering in the translation into Catalan and viceversa. E.g. the Spanish expression *la mesa cuyo propietario es* (the table whose owner is) would correspond to *la taula el propietari del qual és*.

Gender concordance A masculine Spanish term can correspond to a feminine Catalan term, and viceversa. E.g. *la señal* (Spanish fem., the signal) corresponds to *el senyal* (Catalan masc.).

Unknown words Apart from the numbers, there are other words that are not found in the training corpus due to the fact that they appear only at the beginning of the sentence in capital letters, so that the same words written in lower case letters are not translated.

4 Applying improvement techniques

In order to solve some of the problems described in the previous section, three different techniques have been applied, based on the use of the grammatical category of the words, lexical categorisation and direct text processing, respectively.

4.1 Grammatical category-based techniques

Grammatical categories have been successfully implemented in statistical machine translation in order to deal with some problems such as reordering (Crego and Mariño, 2007) and automatic error analysis (Popović and Ney, 2006). The aim is to add the grammatical category (tag) corresponding to the word we are dealing with, so that the statistical model will be able to distinguish the words according to its category and to learn from context.

Homonymy disambiguation

In the translation task, it is common to find two words in the source language with the same spelling and different meaning that correspond to two different words in the target language, which leads to incorrect translations. When equal words in the source language differ from each other by their grammatical category or associated tag (they are homonymous), such tag can be used for disambiguation.

In the case of the Catalan verb *soler*, instead of generating a series of rules to detect whether *sol* and *sols* are verbal conjugations of *soler*, the tag is directly taken from Freeling tool (Carreras et al., 2004).

However, in some cases, the tag information given by the FreeLing tool is not correct, and some additional processing is needed in order to perform the word disambiguation. In the *solo* case, a series of context-based rules have been designed to identify the *solo* adverb from the *solo* adjective in the doubtful cases. The rules are applied over the source language and the corresponding tag is added to the word in question. Thus, a source language sentence such as *venía solo* (he was coming alone) is transformed into *venía solo_<ADJ>*, so that the statistical model will be able to distinguish between both cases.

A similar process is performed in the Catalan possessives: a set of rules has been designed in order to assign a tag indicating the category of the word (adjective or pronoun), and the tags are then implemented in the source language. Some examples of the resulting translations after applying homonymy disambiguation can be found in Table 1.

Soler	(S) La CR **sol** disposar de quatre.
	(T1) La CR ***solo** disponer de cuatro.
	(T2) La CR **suele** disponer de cuatro.
Solo	(S) Era **solo** un niño.
	(T1) Era ***sol** un nen.
	(T2) Només era un nen.
Poss.	(S) Els meus amics no són **els teus**.
	(T1) Mis amigos no están ***tus**.
	(T2) Mis amigos no son **los tuyos**.

Table 1: Examples of correction after homonymy disambiguation.

Pronominal clitics

The pronominal clitics are initially detected and separated from the verb by using the Freeling tool. After translating them, they are combined again with the corresponding verb. In order to solve the errors in this combination process, a set of rules is defined, in which two grammatical aspects are considered: the Spanish accentuation rules and the pronoun-verb combination in Catalan. In Spanish, for instance, the stressed syllable position changes when adding an enclitic pronoun to the verb:

vende + ***lo*** → *véndelo* (sell **it**)

while in Catalan, the accentuation rules are not altered and the pronoun-verb combination is performed by using apostrophes or hyphens:

seguir + ***lo*** → *seguir-**lo*** (follow **it**)
compra + ***el*** → *compra'**l*** (buy **it**)
el + *aixecava* → ***l'***aixecava (lifted **it**)

Apostrophe

A series of rules have been applied in order to fulfil the Catalan apostrophe rules. The basic apostrophe rule states that the singular articles *el, la* and the preposition *de* must be apostrophised when preceding a word that begins with a vowel or an unsounded *h* (in Catalan language the letter *h* is not pronounced):

el + *arbre* → *l'arbre* (the tree)
la + *hora* → *l'hora* (the hour)
de + *eines* → *d'eines* (of tools)

Some exceptions to these rules have also been included:

- The articles and the preposition are not apostrophised when they precede terms beginning with semiconsonantic *i* or *u* (including *hi*, *hu*): *el uombat* (the wombat), *la hiena* (the hyena), *de iogurt* (of yoghurt).
- The feminine article is not apostrophised when precedes a word that begins with atonic *i* or *u* (including *hi* and *hu*): *la universitat* (the university), *la Irene*.
- The feminine article and the preposition are not apostrophised when preceding the negative prefix *a*: *la anormalitat* (the anormality), *de asimètric* (of asimètric).
- *La una* [hora](one o'clock), *la ira* (the wrath), *la host* (the host) and the names of letters(*la e*, *la hac*, *la erra*, etc.) are not apostrophised.

Some examples of clitics and apostrophe correction can be found in Table 2.

Clitics	(S) No quiero ver**te** más por aquí.
	(T1) No vull veure ***et** més per aquí.
	(T2) No vull veure'**t** més per aquí.
Apostr.	(S) **La acepta** hasta el final.
	(T1) ***La accepta** fins al final.
	(T2) **L'accepta** fins al final.

Table 2: Examples of clitics and apostrophe correction.

Capital letters at sentence beginning

It was also seen in section 3 that some of the unknown words appear in the training corpus only in capital letters, since they are found only at the beginning of sentences. In order to solve this problem, all those words that appear at the sentence beginning are changed to lower case words, except for proper nouns, common nouns and adjectives, since common nouns and adjectives could be also proper nouns, and they are usually not found at sentence beginnings. Therefore, those words that appeared only in capital letters will be translated when writing them in lower case. An example of this type of correction can be found in Table 3.

(S) No entenc per què no hi **assisteixes**.
(T1) No entiendo por qué no ***assisteixes**.
(T2) No entiendo por qué no **asistes**.

Table 3: Example of capital letter unknown word correction.

Gender concordance

In order to improve the translation of those words that change the gender between Catalan and Spanish, a tag containing the part-of-speech information has been used. This technique benefits those word sequences that maintain the gender coherence; for instance: *pilota*_FN *verda*_FAdj (where FN is feminine noun and FAdj feminine adjective) will have a higher probability that *pilota*_FN *verd*_MAdj (where MAdj is a masculine adjective), since the tags model will have seen more time the sequence FN-FAdj than the sequence FN-MAdj.

Nevertheless, the tags model will be useful only if the language model (i.e. the tuples included in the training corpus) allows it. Thus, the translation of *senyal*_MN *blanc*_MAdj will remain as *señal*_FN *blanco*_MAdj instead of *señal*_FN *blanca*_FAdj, since the tuple *blanc#blanca* is not contained in the translation model.

Cuyo

In order to solve the problem of the relative pronoun *cuyo*, a preprocessing rule was applied to transform the Spanish structure into a literal translation of the Catalan structure *del qual*; i.e. the sentences containing *cuyo* or some of its other forms (*cuya*, *cuyos*, *cuyas*), were transformed to

sentences containing *del cual* or its corresponding forms (*de la qual*, *de los cuales*, *de las cuales*), so that the alignment was easier, and some translation errors related to this pronoun were avoided.

Table 4 shows some examples of gender concordance and *cuyo* correction.

Gender	(S) Me encantan **las espinacas**.
	(T1) M'encanten ***les espinacs**.
	(T2) M'encanten **els espinacs**.
Cuyo	(S) Un pueblo **cuyo** nombre es largo.
	(T1) Un poble ***amb un nom** és llarg.
	(T2) Un poble el nom **del qual** és llarg.

Table 4: Examples of gender concordance and *cuyo* relative pronoun correction.

4.2 Numbers and time categorisation

As it was seen in section 3, many numeric expressions are not included in the training corpus and they appear as unknown words in the translation process. In order to solve this problem, the numeric expressions are detected in the source language, codified, and generated again in the target language.

In order to detect the numbers in the source language, two issues must be considered: the structure of the numeric expressions (compound words, use of dashes, etc.) and the gender of the number, if applicable. Then, a specific codification is defined in order to maintain the coherence of the detected expression. Numbers like *un/una* (one), *nou* (nine) and *deu* (ten) have not been categorised because they can be related to non numeric expressions.

On the other hand, it was also seen in section 3 that time expressions differ in Catalan and Spanish languages. Since the training corpus contains few examples related to time expressions, it is difficult to learn from context and to obtain correct translations. As in the numbers, time expressions are detected (considering three possible expression structures), codified and generated in the target language. In some cases, where a verb exists, this changes in the translation, so that it becomes necessary to include it in the detection step. In the following Catalan-Spanish example: *són dos quarts de dues* (it's half past one), which is translated into *es la una media*, the verb changes from plural to singular; thus, the verb must also be included in the detected structure.

Some examples of the correction after number and time categorisation can be found in Table 5.

Numbers	(S) L'alliberament de **quatre-cents quaranta-un** presoners.
	(T1) La liberación de ***quatre-cents *quaranta-un** prisioneros.
	(T2) La liberación de **cuatrocientos cuarenta y un** prisioneros.
Hours	(S) Són **tres quarts de vuit**.
	(T1) Son ***tres cuartos de ocho**.
	(T2) Son **las ocho menos cuarto**.

Table 5: Examples of correction after number and time categorisation.

4.3 Text processing

Some of the errors need to be solved by performing a text processing before or after the translation. The geminated *l*, for instance, have been treated before the translation, by normalising the writing of the middle dot. In other cases such as the obligation *tener que* and the conjunctions *y* and *o* have been treated as a postprocessing after the translation. Some examples correction by text processing can be found in Table 6.

Gemin.	(S) Reformat a **Brussel.les**.
l	(T1) Reformado en ***Bruselas. las**.
	(T2) Se ha reformado en **Bruselas**.
Obligat.	(S) Nos lo **tenemos que** creer.
	(T1) Ens ho ***tenim que** creure.
	(T2) Ens ho **hem de** creure.
y/o	(S) Com a Blanes **o** Olot.
	(T1) Como Blanes ***o** Olot.
	(T2) Como Blanes **u** Olot.

Table 6: Some examples of text processing correction.

5 Evaluation

In order to evaluate the final system after applying the grammatical rules and statistical techniques described in the current paper, a test corpus containing the above-mentioned problematic cases was developed. The built corpus contains 636 sentences for each of the source and target languages, where the problems to deal with can be found in a balanced proportion. In addition, an evaluation with a 2000-sentence test extracted from *El Per-*

iódico itself was also performed. The obtained results are shown in Table 7.

	Sent.	ES > CA	CA > ES
Baseline N-II	636	75.91	73.50
Improved N-II		81.35	76.12
Baseline N-II	2000	83.80	83.01
Improved N-II		83.91	83.23

Table 7: BLEU results in both directions of translation.

The results obtained with the 636-sentence test corpus show that the problems we were focusing on are being solved better than in the baseline system. A slight improvement is also observed when using the *El Periódico* test set, although the improvement is not so obvious since the corpus does not contain explicitly the error cases we were dealing with. Additionally, the following points could explain some reasons why the improvement was not higher:

1. The improved translation has an additional knowledge with respect to the corpus. Therefore, some translations from the improved system are correct but differ from the reference while the baseline system outputs the reference as it is. E.g. *EUA està (...)* instead of *Els EUA estan (...)*

2. The CA>ES translation from the improved system contains more words than the CA>ES translation from the baseline system. It must be taken into account that BLEU measures the precision and not the recall.

6 Conclusions

The initial aim of the current paper was to improve an Ngram-based statistical machine system. Once a set of common errors were detected through a human evaluation, a set of techniques based on the used of grammatical category, lexical categorisation and text processing have been applied.

When using an *ad hoc* built test corpus, the results show that the use of grammatical information and the correction of the text as a pre- and postprocessing are useful techniques in order to achieve this goal, as it has been shown in the automatic evaluation: the BLEU of the improved N-II is higher with respect to the baseline system.

A higher performance in terms of BLEU is also reflected in the improved N-II when using a general corpus extracted from *EL Periódico*, although the relative improvement is less than the previous one, since the corpus does not contain explicitly the problems we were tackling in the current paper. Additionally, possible causes for the less improvement observed have been analysed.

References

Carreras, Xavier , Chao, Isaac , Padró, Lluís, and Padró, Muntsa. 2004. FreeLing: An Open-Source Suite of Language Analyzers. *Proceedings of the Conference on Language Resources and Evaluation*, Lisboa.

Crego, Josep M. and Mariño, José B. 2007. Improving SMT by coupling reordering and decoding. *Machine Translation*, 20:3:199–215.

Mariño, José B. , Banchs, Rafael E. , Crego, Josep M. , de Gispert, Adrià , Lambert, Patrick , Fonollosa, J.A.R. and Costa-jussà, Marta R. 2006. N-gram Based Machine Translation. *Computational Linguistics*, 32:4:527–549.

Niessen, S., Ney, H. 2000. Improving SMT quality with morpho-syntactic analysis. *Proceedings of the International conference on Computational Linguistics*, Saarbrücken, Germany.

Och, Franz Josef 2003. Minimum Error Rate Training in Statistical Machine Translation. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, Sapporo, Japan. 160–167.

Popović, M., Ney, H. 2004. Towards the Use of Word Stems and Suffixes for Statistical Machine Translation. *Proceedings of International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Popović, M., Ney, H. 2006. POS-based Word Reorderings for Statistical Machine Translation. *Proceedings of International Conference on Language Resources and Evaluation*, Genoa, Italy.

Popović, M., de Gispert, A., Gupta, D., Lambert, P., Ney, H., Mariño, J.B. y Banchs, R. 2006. Morpho-syntactic Information for Automatic Error Analysis of Statistical Machine Translation Output. *Proceedings of the HLT/NAACL Workshop on Statistical Machine Translation*, New York.

Use of Rich Linguistic Information to Translate Prepositions and Grammatical Cases to Basque

Eneko Agirre, Aitziber Atutxa, Gorka Labaka, Mikel Lersundi,
Aingeru Mayor, Kepa Sarasola
IXA Group. University of the Basque Country
aingeru@ehu.es

Abstract

This paper presents three successful techniques to translate prepositions heading verbal complements by means of rich linguistic information, in the context of a rule-based Machine Translation system for an agglutinative language with scarce resources. This information comes in the form of lexicalized syntactic dependency triples, verb subcategorization and manually coded selection rules based on lexical, syntactic and semantic information. The first two resources have been automatically extracted from monolingual corpora. The results obtained using a new evaluation methodology show that all proposed techniques improve precision over the baselines, including a translation dictionary compiled from an aligned corpus, and a state-of-the-art statistical Machine Translation system. The results also show that linguistic information in all three techniques are complementary, and that a combination of them obtains the best F-score results overall.

1 Introduction

Since the first Machine Translation (MT) systems up to today's, performing well the translation of the prepositions is relevant for any MT system; Japkowicz and Wiebe (1991) claimed that doing it correctly is difficult because prepositions cannot be translated in a systematic or coherent way. Koehn (2003) remarked the importance of the correct translation of prepositions and he also reported that the main reason for noun phrase (NP) and prepositional phrase (PP) mistranslations consists of choosing wrong leading preposition.

Translation of prepositions is even more complex when the verb phrase and prepositional phrase structures differ widely in the languages involved in translation (Naskar and Bandyopadhyayn, 2006). This is what happens when translating from Spanish or English into Basque.

This paper explores the problem of translating prepositions heading verbal complements into target language equivalents. Although we focus on Spanish to Basque translation, the evaluation methodology and techniques can be applied to other language pairs. In Basque syntactic functions like subject, object and indirect objects are marked by case-suffixes. In this work postpositions and grammatical cases have been homogeneously treated, therefore it covers not only the translation of Spanish prepositions, but also how to choose the correct grammatical case corresponding to Spanish subjects, objects and indirect objects. Note that in most of the cases Spanish subjects and objects are not marked by any surface word or special case marking. Thus, besides the Spanish prepositions, we also explore the translation of the *zero preposition* corresponding to the grammatical cases of subject and object.

Given an existing open-source rule-based machine translation (RBMT) system called *Matxin* (Mayor, 2007; Alegria et al., 2007), we propose and evaluate three different techniques for translating Spanish prepositions and syntactic functions into Basque. These techniques use rich linguistic information like verb/postposition[1]/head-word dependency triples, verb subcategorization and manually coded selection rules based on lexical, syn-

[1]When we use here the word postposition, we would like to refer to grammatical cases and postpositions

Proceedings of the 13th Annual Conference of the EAMT, pages 58–65,
Barcelona, May 2009

tactic and semantic information. While the latter rules have been coded manually, the first two resources have been automatically extracted from monolingual corpora.

One important contribution of this paper is the evaluation methodology. Previous work (Husain et al., 2007; Gustavii, 2005) on preposition translation measured only accuracy gains with respect to simple baselines, and focused on small sets of frequent prepositions. Our methodology measures both precision and recall over all prepositions occurring in a small corpus of randomly chosen sentences. Once the evaluation corpus has been compiled, the evaluation is fully automatic.

The results of this paper shows that all proposed techniques improve over the baselines, including a translation dictionary compiled from an aligned corpus, and over a full-fledged statistical machine translation (SMT) system. The results also show that the linguistic information in all three techniques is complementary, and a combination of them obtains the best results overall.

In the next section of this paper we describe the RBMT system used, followed by a small review of related work on preposition translation. We then present the linguistic knowledge used. Section 5 presents the different baselines and techniques to translate prepositions. Our evaluation methodology is proposed in Section 6, which is followed by Section 7 with the results. Finally, Section 8 is devoted to conclusions and future work.

2 Preposition translation in RBMT

The last decade has seen the raise of SMT techniques, and less research on rule-based techniques. Nevertheless, translation involving a less-resourced language poses serious difficulties for SMT, specially caused by the smaller size of parallel corpora. Morphologically-rich languages have also been proved to be difficult for SMT, as shown in (Koehn and Monz, 2006), where SMT systems lag well behind commercial RBMT systems. At present, domain-specific translation memories for Basque are no bigger than two or three million words, much smaller than corpora used for other languages (the Europarl parallel corpus, for instance, has ca. 30 Mwords). Having limited digital resources, the rule-based approach is suitable for the development of an MT system for Basque, along with a focus on the enhancement of the core RBMT system with statistical and linguistic information.

The freely available open-source *Matxin* system is the first MT system available for Basque. It is a rule-based transfer system based on deep syntactic analysis. which currently translates from Spanish into Basque, and is currently being adapted to the English-Basque pair. The current development status shows that it is useful for content assimilation, for text understanding indeed, but that it is not yet suitable for unrestricted use in text dissemination.

Matxin has been evaluated and compared with the state-of-the-art corpus-based *Matrex* MT system (Stroppa et al., 2006; Labaka, 2007) translating from Spanish to Basque. The evaluation was performed using the *edit-distance* metric (Przybocki et al., 2006), based on the *HTER* (human-targeted translation edit rate) presented in (Snover et al., 2006), and the comparative results have shown that *Matxin* performs significantly better: 43.60 vs. 57.97 in the parallel corpus where *Matrex* was trained, and 40.41 vs. 71.87 in an out-of-domain corpus.

The preposition translation module of *Matxin* is located in the structural transfer phase and uses the information carried over from the syntactic analysis and lexical transfer modules. The system currently uses *Freeling* analyzer for Spanish (Atserias et al., 2006). The output of the preposition translation module is later used in subsequent modules in the structural transfer and generation phases. Note that errors from previous modules affect the quality of the preposition translation phase, and this makes the separate evaluation of preposition translation a difficult task. We will get back to this problem in Section 6.

3 Related work

Koehn (2003) envisages MT as a divide and conquer task where improving NP/PP translation will carry an improvement of the whole system. That study concluded that the main source of re-ranking errors in NP/PPs translation was the inability to correctly predict the phrase start (preposition or determiner) without context; it can sometimes only be resolved when the English verb is chosen and its subcategorization is known.

There are two main approaches to disambiguate prepositions (Mamidi, 2004; Alam, 2004; Trujillo, 1992): context based (used in transfer systems and more suitable for languages that are structurally different) and concept based (used in interlingua

systems and more suitable for languages which are very close). Most of the systems are context based and they use transfer rules given with semantic information for the nouns which are head and complement of the preposition.

(Miller, 2000) argued that statistical models for preposition selection must take into account not only affinities between verbs and prepositions, but affinities between prepositions and nouns functioning as their complement as well.

(Husain et al., 2007) describes an approach to automatically select from two Indian languages the appropriate lexical correspondence of English simple prepositions. They use a set of rules that deal with syntactic and lexical-semantic constraints on the head and complement of the preposition. The results showed relative improvements greater than 20% in precision when compared to the default sense, but the experiments were conducted with just 6 high frequency prepositions. The algorithm was tested on 100 sentences for each preposition The input to the implemented system had been manually checked and corrected to make sure that there were no errors in the PP attachment given by the parser and no mistakes in phrasal verb identification.

(Naskar and Bandyopadhyayn, 2006) describes how the prepositions are handled in an English-Bengali MT system. As in Basque, there is no concept of preposition in Bengali. English prepositions are translated using inflections and/or postpositional words. The choice of the appropriate inflection depends on the spelling of the complement of the preposition and the choice of the postpositional word depends on its semantic information, obtained from the *WordNet*. They don't report any evaluation.

(Gustavii, 2005) corrected the preposition translations using a TBL classifier. She used aligned bilingual corpus data to infer her classifiers. Her evaluation is performed giving translation accuracy for only the six most frequent prepositions in the training corpus. She used a subset of 3 million tokens of the Swedish-English Europarl corpus, 90% for training and 10% for testing. The relative total improvement is of 12,45% (75,5% accuracy for the baseline and 84.9% for her system). However the applicability of the strategy is limited to relatively similar languages, as the ones of that study (Swedish and English). In fact the system avoids inducing rules where a preposition should be changed to some other part-of-speech, or where it should be completely removed. So this approach is not useful to translate from Spanish to Basque.

Freq.	Transitivity	Postpositions
4289.78	transitive	ABS,ERG
1534.24	intransitive	ABS
975.31	transitive	ABS,ERG,INE
476.70	intransitive	ABS,INE
166.68	transitive	ABS,ERG,INS

Table 1: Subcategorization for verb *ikusi* (*to see*).

4 Acquisition of rich linguistic information from corpus

Before showing our specific techniques for preposition translation, we briefly present the linguistic resources used, and how they were automatically acquired from Basque monolingual corpora.

4.1 Verb subcategorization

One of the information sources used for this experiment was an already existing subcategorization dictionary, initially built with the purpose of making attachment decisions for a shallow parser on its way to full parsing (Atutxa, forthcoming). For each of the 2,571 verbs this dictionary lists information about possible postposition and grammatical case combinations, transitivity, and estimated frequency of each combination. Table 1 shows the most frequent patterns in the dictionary entry for verb *ikusi* (*to see*), including estimated frequency, transitivity and postpositions (including grammatical cases) [2].

This dictionary was automatically built from raw corpora, comprising a compilation of 18 months of news from *Euskaldunon Egunkaria* (a newspaper written in Basque). The size of the corpus is around 780,000 sentences, approximately 10 Mwords. From the 5,572 different verb lemmas in the corpus, the subcategorization dictionary was compiled for the 2,751 verbs occurring at least 10 times.

The corpus was parsed by a chunker (Aduriz et al., 2004) which includes both named-entity and multiword recognition. The chunker uses a small grammar to identify heads, postpositions and verb attachments of NPs and PPs. The grammar was developed based on the fact that Basque is a head

[2] ABS : absolutive case (can be subject or object depending on transitivity). ERG : ergative (subject with transitive verbs). INE : inesive. INS : instrumental. DAT : dative. ALA : allative.

final language and it includes a distance feature as well. Phrases were correctly attached to the verb with a precision of 0,78. Note that the auxiliary verb in Basque allows to unambiguously determine the transitivity of the main verb. Given the fact that Basque is a three-way pro-drop language (subject, object and indirect object can be elided), cases of elided arguments were recovered from the auxiliary verb in most of the cases. The only exception were unergative verbs (e.g. *lo egin* – to sleep), which incorporate the missing argument. Statistical thresholds were used to reduce the errors caused by unergative verbs and wrong verb attachment decisions.

4.2 Verb/postposition/head-word dependency triples

Verbal subcategorization can be also modeled using attested (verb, dependency, head word) triples. The postposition can be used as the type of the dependency. In contrast to the subcategorization dictionary, and given that the headword is also kept, these triples are bound to be more sparse. Due to sparseness, the statistical threshold used for subcategorization acquisition proved to be ineffective, and it was devised an alternative acquisition method.

Only dependencies from the preverbal position of each clause were extracted. This position is the focus position of Basque, and the probability that a phrase at this position is attached to the verb just behind is quite high (up to 0.93 precision). Given the fact that Basque is a free word order language, and provided it is used a large enough corpus, it can be expected all arguments of a given verb to appear at the preverbal position in some attested sentence. This way, most of the potential arguments of a verb would be attested in the preverbal position, and therefore be captured as licit arguments of the verb. Table 2 shows the top triples for verb *ikusi* (*to see*). Attested headwords in the example include also elided pronouns and named-entities (of types PERSON, LOCATION, ORGANIZATION).

Freq.	Postposition	Head word
70	ERG	PRONOUN
36	ABS	PRONOUN
30	ERG	PERSON
16	INE	LOCATION
13	ABS	talde (group)
11	ABS	LOCATION
9	ABS	ORGANIZATION
9	ABS	partidu (match)

Table 2: Dependency triples for verb *ikusi*.

5 Strategies for preposition translation

In this section we present both the dictionary and aligned corpora baselines, alongside our three methods to translate prepositions: a context based approach using manually coded selection rules, and the use of subcategorization information or dependency triples to disambiguate the prepositions heading verbal complements.

5.1 Baselines

The baseline dictionary uses the preposition translations in the *Elhuyar* dictionary (Elhuyar, 2000), the most popular Spanish-Basque dictionary. The first postposition is taken as the preferred translation.

The aligned corpora baseline was constructed applying Giza++ (Koehn et al., 2003) to the *Consumer* magazine parallel corpus (Alcazar, 2006). This corpus contains 60,000 parallel sentences in Spanish (1.3 Mwords) and Basque (1 Mwords). The Basque part of the corpus was morphologically analyzed and segmented, i.e. word forms were split into their lemma and postposition (e.g.: *etxetik* (from the house) → *etxe* (the house) + *tik* (from)). After preprocessing the Basque sentences, we aligned the text automatically and extracted for each Spanish preposition its most frequent corresponding Basque postposition. This alignment technique proved to be superior to word-base alignment (Agirre et al., 2006). For a given Spanish preposition, the most frequent alignment was chosen as its Basque translation.

Note that these techniques do not tackle the translation of subject and object *zero prepositions* into Basque postpositions. In both baselines prepositions are always translated in the same way, irrespective of the context of occurrence of the preposition.

5.2 Selection rules

The preposition dictionary used as baseline above contains 351 Spanish prepositions (18 simple and 333 compound) plus what we call *zero preposition* for subject and object, and the possible Basque postpositions (462 in total) into which they can be translated. We have manually coded 89 selection rules to select the appropriate equivalent for the ambiguous prepositions.

Prep.	Postpos.	Rule
a	INE	./[@nounPOS='Zm']
a	DAT	-
a	ABS	-
a	ALA	./[@si='cc']

Table 3: Rule for the Spanish preposition *a*.

The rules contain lexical, syntactic and semantic information about the parent of the PP, and about the words in the PP (mainly the head).

Selection rules select or discard possible postpositions for one preposition, and can thus return, in general, more than one postposition. In the case of multiple suggestions, another method would be used to choose among those returned by the selection rules.

For example, given the sentence *Los venden a tres euros* (They sell them for three euros), the possible translations for the preposition *a* are the cases INE, DAT, ABS and ALA, as we can see in Table 3[3]. The rule that selects INE is applied because the *part-of-speech* of the head of the prepositional phrase is *Zm* and thus the selected translation will be INE: *Hiru eurotan saltzen dituzte*.

5.3 Verb subcategorization

Given a source sentence, the system accesses its syntactic analysis (provided by *Freeling* Spanish parser) and retrieves the verbs and a list with their dependent NPs and PPs. We process each verb in turn. For each of the NPs and PPs, the dictionary is used to retrieve all possible translations of the prepositions, building a data structure that contains the main verb and a list of potential translations for each of its dependent NPs and PPs. We also retrieve the translation of the main verb as produced by the lexical selection modules of *Matxin*. The algorithm then examines the subcategorization patterns of the translation of the verb, starting from the most frequent one, until it finds a pattern that matches the aforementioned data-structure.

For instance, given a source sentence like *yo he visto a tu madre* (I have seen your mother), we retrieve the main verb (*visto* - seen) and two dependents: the subject NP (*yo* – I) and the direct object which in Spanish uses the preposition *a* (*a tu madre* – your mother). The possible translations for the zero preposition are ABS, ERG and INE. The possible translations for *a* are ABS, DAT, ALA and INE. Given the translation of the verb, *ikusi* as suggested by *Matxin*, we can now access its subcategorization patterns from the dictionary as described in Section 4.1. The most frequent pattern for *ikusi* is (transitive, ABS,ERG), as shown in Table 1. As this subcategorization frame matches the example (ERG for *yo* and ABS for *a tu madre*) ERG and ABS grammatical cases are selected as translation in Basque. This information would be passed onto the generation module of *Matxin*.

5.4 Verb/postposition/head-word dependency triples

The algorithm in this case is very similar to that used in the subcategorization method. For each verb in the source sentence, we generate a data structure with the translation of the verb, and the list of dependent NPs and PPs, with the possible translation postpositions for each. Here we also add the translations into Basque of all heads of NPs and PPs.

Contrary to subcategorization, we treat each dependent NP and PP independently, one at a turn, choosing the most frequent dependency triple which matches the translation of the verb, one of the translations of the postposition and the translation of the noun. In other words, we choose the postposition which occurs first in the triples for this verb and head-word combination.

We will illustrate this example with a different example. Given the source sentence *El se conecta a Internet* (He connects to the Internet), we focus on the translation of the *a* preposition. *Matxin* translates *conecta* as *konektatu* and *Internet* as *Internet*. Given the set of possible translations for *a* (ABS, DAT, ALA and INE), the list of triples containing *konektatu* and *Internet* is checked, and the ALA postposition is chosen as the most frequent one for those.

5.5 Combination of techniques

Given a set of single techniques for preposition translation, we can combine them in several ways. Most of the techniques above have partial recall (i.e. they sometimes are not able to choose a single best translation), due mainly to sparse data problems. We therefore decided to combine them in cascade, one after the other, disambiguating in each step the prepositions which had not been translated in the previous one. We tried several combinations, as will be shown in the following

[3] *si*: syntactic information. *cc* circumstancial complement. *Zm*: tag for currency.

Phrase	Preposition	Postposition
El mensaje	-	ABS
por correo	por	INS
a su amiga	a	DAT

Table 4: An example of the gold standard.

section, but the cascade always orders the techniques according to their precision in the test set.

6 Evaluation framework

We ruled out the use of *Bleu* because, as pointed in (Callison-Burch et al., 2006), it cannot be always used to identify the improvements of the aspects of the translation. In our case, it is impossible to establish how much the *Bleu* score should rise or drop to detect significant improvements in the translation of prepositions.

We designed the evaluation framework in order to provide automatically both precision and recall for all prepositions. To create the gold standard, we selected 300 sentences at random from a parallel corpus of newspapers and technical reports. As our evaluation had to isolate the preposition translation task, the output of previous modules in the MT engine for each sentence was examined and if there was any mistake that affected the preposition translation (e.g. in the source text analysis or in the verb transfer), we discarded the sentence. In the remaining 54 sentences there were 80 Spanish prepositions and 81 syntactic functions (subject, direct object and indirect objects) to translate.

Table 4 shows an example of the gold standard. For the sentence *El mensaje ha sido enviado por correo a su amiga* (The message has been sent by mail to her friend) we coded the correct postposition for the prepositions (included the *zero preposition* in subject) of these three phrases: *El mensaje* (The message), *por correo* (by mail), *a su amiga* (to her friend).

7 Evaluation results

Table 5 shows, for each strategy, the number of correctly translated postpositions and the total number of postpositons translated (both correctly and incorrectly), alongside the overall number of cases in the test case. Precision, recall and F-score (actually, F_1) are also included. Significance ranges for F-score have been computed using bootstrap resampling for 95% confidence. Given the small size of the dataset, the significance ranges are quite large, over 5 percentage points on all cases.

The first set of rows shows the results for the baselines. We can see that the dictionary performs better than the translations coming from the aligned corpus, which was an unexpected finding. Both baselines return a translation in all cases, and have recall identical to the precision.

The second set of rows describes the performance of each of the techniques proposed in this paper. The manually coded selection rules method has the highest precision, but it scores second in recall and F-score. Subcategorization obtains the lowest precision from the three techniques, but the best recall and F-score. The precision of all of our techniques improves over the baselines, but, due to the fact that they don't always provide a translation, recall and the F-score are lower.

Regarding combination, the third set of rows presents several cascades of techniques. Combining single techniques with the first sense baseline basically provides full coverage and improves recall, providing non-significant improvements on F-score for rules and triples, and statistically significant improvement for subcategorization. The pairwise combination of two techniques gets good precision, but not full coverage, and F-score is similar to the 1st sense baseline. On the same set of results the cascade of all three methods is reported to have very high precision and recall.

The last four rows report the results for pairwise and three-wise combinations of the techniques with the 1st sense baseline. The improvement is consistent in all combinations, and the best result is for the combination of all.

Given the small number of examples only a few performance differences are statistically significant. Below we list the pairs of results (among those which have full coverage, i.e. those using 1st sense) that are statistically significant:

1st sense $<$ a+b+c+1st
a+1st $<$ a+b+c+1st
b+1st $<$ a+b+c+1st

Regarding the comparison among techniques, and although the differences are not statistically significant, the combinations that use subcategorization are the ones performing best, and it is always the single technique which improves most in each combination class. This is further enforced by the fact that a+1st and b+1st perform significantly worse than a+b+c+1st, while the difference

	Correct	Translated	Overall	Precision	Recall	F-score Signif.
Baselines						
Dictionary	109	161	161	**67.70%**	**67.70%**	**67.70%** ±6.26
Alignment Dict.	101	161	161	62.73%	62.73%	62.73% ±5.98
Techniques						
Rules (a)	73	83	161	**87.95%**	45.34%	59.84% ±6.73
Triples (b)	54	62	161	87.10%	33.54%	48.43% ±7.40
Subcat (c)	84	107	161	78.50%	**52.17%**	**62.69%** ±6.78
Combinations						
a+1st	110	161	161	68.32%	68.32%	68.32% ±6.64
b+1st	111	161	161	68.94%	68.94%	68.94% ±6.30
c+1st	116	161	161	72.05%	72.05%	72.05% ±5.42
a+b	87	98	161	**88.78%**	54.04%	67.18% ±6.09
b+c	89	112	161	79.46%	55.28%	65.20% ±6.41
a+c	99	124	161	79.84%	61.49%	69.47% ±6.11
a+b+c	103	125	161	82.40%	63.98%	72.03% ±5.48
a+b+1st	115	161	161	71.43%	71.43%	71.43% ±5.92
b+c+1st	118	161	161	73.29%	73.29%	73.29% ±5.91
a+c+1st	117	161	161	72.67%	72.67%	72.67% ±5.68
a+b+c+1st	121	161	161	75.16%	**75.16%**	**75.16%** ±5.70

Table 5: Overall results of baselines, single techniques and combinations.

	Correct	Translated	Overall	Precision	Recall	F-score Signif.
$\text{SMT}_{wordforms}$	60	161	161	37.27%	37.27%	37.27% ±6.84
$\text{SMT}_{segmented}$	82	149	161	55.03%	50.93%	52.90% ±6.35

Table 6: Results for SMT systems trained with word forms and segmented words

between c+1st and a+b+c+1st is not significant.

Table 6 shows the results obtained by two state-of-the-art full-fledged SMT systems, one of them was trained using Basque word forms for alignment, and the other using Basque segmented words (see Section 5.1). The whole sentences were translated and then the postpositions related to the translated phrases were compared with the gold standard. Their results are clearly lower than those obtained with each of the three simple strategies or any of their combinations.

8 Conclusions and future work

In this work, three techniques that use rich linguistic information to translate grammatical cases and prepositions heading verbal complements have been implemented and successfully evaluated in the context of an RBMT system for an agglutinative language with scarce resources. They are based on verb/postposition/head-word dependency triples, verb subcategorization and manually coded selection rules based on lexical, syntactic and semantic information. The first two resources have been automatically extracted from monolingual corpus, that obviously is easier to collect than parallel corpus. As traslation involving a less resourced language poses serious dificulties for pure SMT, we think these two techniques based on monolingual corpus statistics are opening new ways to integrate rule-based and statistical-based techniques in MT languages with fewer digital resources.

A new methodology of evaluation has been designed. It allows to automatically measure precision and recall against a gold standard. Even if our test corpus is not very large, it is comparable with those used in related work, and the F-scores show that some of the improvements are statistically significant.

The proposed techniques improve precision over the baselines, including a translation dictionary compiled from an aligned corpus, and over a full-fledged SMT system. The results also show that the linguistic information in all three techniques is complementary, and a combination of them obtains the best results overall.

In the near future we plan to collect larger linguistic resources to obtain better information on verb subcategorization and verb/postposition/head-word triples, so we could improve our present results. We also plan to enlarge the gold standard and to evaluate the relevance of our techniques in overall translation quality, using the edit-distance metric (Przybocki et al., 2006). We would also like to use the output of SMT systems in the combined system.

Acknowledgments

This research was supported in part by the Spanish Ministry of Education and Science (OpenMT: Open Source Machine Translation using hybrid methods, TIN2006-15307-C03-01; RICOTERM-3, HUM2007-65966.CO2-02) and the Regional Branch of the Basque Government (AnHITZ 2006: Language Technologies for Multilingual Interaction in Intelligent Environments, IE06-185). Gorka Labaka is supported by a PhD grant from the Basque Government (grant code, BFI05.326). Consumer corpus has been kindly supplied by Asier Alcázar from the University of Missouri-Columbia and by Eroski Fundazioa.

References

E. Agirre, A. Díaz de Ilarraza, G. Labaka and K. Sarasola. 2006. *Uso de información morfológica en el alineamiento Español-Euskara*. XXII Congreso de la SEPLN.

Y. S. Alam. 2004. *Decision Trees for Sense Disambiguation of Prepositions: Case of Over*. HLT-NAACL 2004: Workshop on Computational Lexical Semantics . Boston, Massachusetts, USA. ACL.

A. Alcázar. 2006. *Towards linguistically searchable text*. Proceedings of BIDE 2005. Bilbao.

I. Alegria, A. Díaz de Ilarraza, G. Labaka, M. Lersundi, A. Mayor and K. Sarasola. 2007. *Transfer-based MT from Spanish into Basque: reusability, standardization and open source*. LNCS 4394. 374-384. Cicling 2007.

I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Díaz de Ilarraza, K. Gojenola, M. Oronoz and L. Uría. 2004. *A Cascaded Syntactic Analyser for Basque.* In Gelbukh, A (ed.) Computational Linguistics and Intelligent Text Processing. Springer LNCS 2945.

J. Atserias, B. Casas, E. Comelles, M. González, L. Padró and M. Padró. 2006. *FreeLing 1.3: Syntactic and semantic services in an open-source NLP library*. Proceedings of the 5th LREC (2006). Genova. Italia.

C. Callison-Burch, M. Osborne and P. Koehn. 2006. *Re-evaluating the role of BLEU in Machine Translation Research*. Proceedings of EACL-2006.

Elhuyar 2000. *Elhuyar Hiztegia*. Published by Elhuyar Hizkuntz Zerbitzuak.

E. Gustavii. 2005. *Target language preposition selection - an experiment with transformation based learning and aligned bilingual data*. Proceedings of the 10th EAMT conference. May 2005. Budapest.

S. Husain, D.M. Sharma and M. Reddy. 2007. *Simple preposition correspondence: a problem in English to Indian language machine translation*. Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions. Prague, Czech Republic. 28 June 2007. pp.51-58.

N. Japkowicz and J. Wiebe. 1991. *A System for Translating Locative Prepositions from English into French*. Proceedings of the Meeting of the ACL. 153-160.

P. Koehn. 2003. *Noun Phrase Translation*. PhD. Thesis.University of Southern California.

P. Koehn, F. Och, and D. Marcu (2003). *Statistical phrase based translation*. In Proceedings of HLT-NAACL 2003, pp. 48-54, Edmonton, Canada.

P. Koehn and C. Monz. 2006. *Manual and Automatic Evaluation of Machine Translation between European Languages*. Proceedings of the Workshop on SMT. ACL. June 2006. New York City. pp. 102–121.

G. Labaka, N. Stroppa, A. Way and K. Sarasola. 2007. *Comparing Rule-Based and Data-Driven Approaches to Spanish-to-Basque MT*. Proceedings of the MT-Summit XI. Copenhagen.

R. Mamidi. 2004. *Disambiguating Prepositions for Machine Translation using Lexical Semantic Resources*. Proceedings of the 'National Seminar on Theoretical and Applied Aspects of Lexical Semantics' organized by Centre of Advanced Study in Linguistics. Hyderabad.

A. Mayor. 2007. *Matxin: Erregeletan oinarritutako itzulpen automatikoko sistema baten eraikuntza estaldura handiko baliabide linguistikoak berrerabiliz*. PhD. Thesis. (In Basque). University of the Basque Country.

K. Miller. 2000. *The lexical choice of prepositions in machine translation*. PhD. Thesis. Upenn.

S.K. Naskar and S. Bandyopadhyayn. 2006. *Handling of Prepositions in English to Bengali Machine Translation*. Proceedings of the EACL workshop on Prepositions, Hyderabad.

M. Przybocki, G. Sanders and A. Le. 2006. *Edit distance: a metric for Machine Translation evaluation*. Proceedings of the LREC-2006. Genoa, Italy.

M. Snover, B Dorr, R. Schwartz, L. Micciulla and J. Makhoul. 2006. *A study of translation edit rate with targeted human annotation*. Proceedings of the Association for Machine Translation in the Americas..

N. Stroppa, D. Groves, A. Way and K. Sarasola. 2006. *Example-Based Machine Translation of the Basque Language*. Proceedings of the 7th conference of the AMTA. pp.232–241. Boston.

A. Trujillo. 1992. *Locations in the Machine Translation of Prepositional Phrases*. Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in MT of Natural Languages. Montreal, Canada.

Gappy Translation Units under Left-to-Right SMT Decoding

Josep M. Crego
Spoken Language Processing Group
LIMSI-CNRS, BP 133
91430 Orsay cedex, France
jmcrego@limsi.fr

François Yvon
Univ Paris-Sud 11
LIMSI-CNRS, BP 133
91430 Orsay cedex, France
yvon@limsi.fr

Abstract

This paper presents an extension for a bilingual n-gram statistical machine translation (SMT) system based on allowing translation units with gaps. Our gappy translation units can be seen as a first step towards introducing hierarchical units similar to those employed in hierarchical MT systems. Our goal is double. On the one hand we aim at capturing the benefits of the higher generalization power shown by hierarchical systems. On the other hand, we want to avoid the computational burden of decoding based on parsing techniques, which among other drawbacks, make difficult the introduction of the required target language model costs.

Our experiments show slight but consistent improvements for Chinese-to-English machine translation. Accuracy results are competitive with those achieved by a state-of-the-art phrase-based system.

1 Introduction

Work in SMT has evolved from the traditional word-based (Brown et al., 1993) to the current phrase-based (Och et al., 1999; Zens et al., 2002; Koehn et al., 2003) and hierarchical-based (Melamed, 2004; Chiang, 2007) translation models. Phrase-based and hierarchical systems are also characterized by the underlying formal device employed to produce translations (Knight, 2008): *finite-state transducers* (FST) on the one hand, and *tree transducers* (TT) on the other hand, specified respectively by rational and context-free grammars, thus implying clear differences in generative power.

A thorough comparison between phrase-based and hierarchical MT can be read in (Zollmann et al., 2008), concluding that hierarchical models slightly outperform phrase-based models under "sufficiently non-monotonic language pairs". One of the reasons for the gap in performance seems to be the ability to generalize using non-terminal categories beyond the strictly lexicalized knowledge represented in phrase-based models.

An illustrative example is given below. It consist of the translation from English to French of negative verb phrases, which yields the alignment of *don't* $X \rightsquigarrow$ *ne* X *pas*, where X could be replaced by almost any finite verb. In this example, the English token *don't* is translated into the French non-contiguous words *ne* and *pas* [1].

The right translation can only be achieved under phrase-based systems, if X (say *want*) has been seen in training next to *don't*, yielding the translation unit:

$$don't\ want\ :\ ne\ veux\ pas$$

In contrast, under hierarchical systems, it is possible to obtain the right generalization, decomposing the previous pattern as:

$$X \rightarrow don't\ Y : ne\ Y\ pas$$
$$Y \rightarrow want : veux$$

[1] This example is only used for illustrative purposes. The contracted form *don't* is not a real issue as most tokenizers split the form as *do not*, thus solving the alignment problem.

Proceedings of the 13th Annual Conference of the EAMT, pages 66–73,
Barcelona, May 2009

This ability to capture better generalization comes at a double price: translation as parsing is typically cubic with respect to the source sentence length; furthermore, in this formalism, target constituent are no longer produced monotonically from left-to-right, thus rendering the application of the language model score difficult (Chiang, 2007).

This example also suggests that hierarchical rules tend to be less sparse, given that the holistic unit in the phrase-based (PB) model is divided into two smaller, more reusable, rules. Notice that, in this specific case, the rich morphology of French verbs increases the sparseness problem of phrase-based translation units. Finally, by using discontinuous patterns, hierarchical translation models can capture large span (bilingual) dependencies.

Other than modeling discontinuous constituents, a major difference between FST- and CFG-based approaches to translation, has to do with the size of the search space, or more precisely with the kind of pruning that takes place to make the search feasible.

As previously outlined, when considering the use of translation units with gaps under the left-to-right decoding approach, the main difficulty arises motivated by the appearance of discontinuities in the output side. In this work, we make use of an input word lattice to naturally avoid this problem, allowing to monotonically compose translation.

Related Work

We follow the work in (Simard et al., 2005), which, to the best of our knowledge is the first MT system that within a left-to-right decoding approach, introduces the idea of phrases with gaps. A main limitation of their work arised from the difficulties of left-to-right decoders to handle gaps in the target side, again because of the non-monotonic generation of the target. Such gaps are to be filled in further steps of the search, thus, increasing the complexity of decoding and at the same time that hindering the use of the target language model.

Such translation units are more naturally used under systems employing parsing techniques to perform the search (hierarchical MT). Different kind of hierarchical translation units have been proposed, which mostly differ from the level of syntactical information they use. We mainly differentiate here between translation units that are formally syntax-based, like those appearing in (Chiang, 2007), which employ non-terminal categories without linguistic motivation, working as placeholders to be filled by words in further translation steps; and hierarchical units that are more linguistically motivated, as in (Zollmann and Venugopal, 2006).

More recently, (Watanabe et al., 2006) presents a hierarchical system in which the target sentence is generated in left-to-right order, thus enabling a straightforward integration of the n-gram language models during search. The authors employ a top-down strategy to parse the foreign language side, using a synchronous grammar having a GNF[2]-like structure. This means that the target side body of each translation rule takes the form $b\beta$, where b is a string of terminal symbols and β a (possibly empty) string of non-terminals. This ensures that the target is built monotonously. (Venugopal et al., 2007) present a hierarchical system that derives translations in two steps, so as to mitigate the computational impact resulting from the intersection of a probabilistic synchronous CFG and and the n-gram language model. Firstly, a CYK-style decoding considering first-best chart item approximations is used to generate an hypergraph of target language derivations. In the second step, a detailed exploration of the previous hypergraph is performed. The language model is used to drive the second step search process and to recover from search errors made during the first step.

Our work differs from theirs crucially in that our system employs a different set of translation structures (units), and because our decoder follows strictly the FST-based approach.

The remaining of this paper is organized as follows. In Section 2, we outline the n-gram-based approach used in the rest of this work. Sections 3 and 3.2 detail the use of translation units with gaps in a left-to-right decoding approach. Translation accuracy results are reported for the Chinese-English language pair in section 4. Finally, in section 5, we draw conclusions and outline further work.

[2]Greibach Normal Form

2 N-gram-based SMT

The baseline translation system described in this paper implements a log-linear combination of several models. In contrast to standard phrase-based approaches (Koehn et al., 2003), the translation model is expressed in *tuples* (instead of phrases), and is estimated as an N-gram language model over such units. It actually defines a joint probability between the language pairs under consideration (Mariño et al., 2006).

We have reimplemented the decoder described in (Crego and Mariño, 2007a), that we have extended to decode input lattices. At decoding time, only those reordering hypotheses encoded in the word lattice are to be examined. Reordering hypotheses are introduced following a set of reordering rules automatically learned from the bi-text corpus word-to-word alignments. Hence, reordering rules are applied on top of the source sentences to be translated.

More formally, given a source sentence, f, in the form of a linear word automaton, and N optional reordering rules to be applied on the given sentence in the form of string transducers (τ_i), the resulting lattice containing reordering hypotheses, f^*, is obtained by the sequential composition of FSTs, as:

$$f^* = \tau_N \circ \tau_{N-1} \cdots \circ \cdots \tau_1 \circ f$$

where $\circ$ denotes the composition operation.

Note that the sequence of FSTs (reordering rules) is sorted according to the length of the left-hand side (LHS) of the rule. More specific rules, having a larger LHS, are applied (composed) first, in order to ensure the recursive application of the rules. Hence, some paths are obtained by applying reordering on top of already reordered paths. Figure 1 illustrates an example where two reordering rules: $abc \rightsquigarrow cab$ (τ_1) and $ab \rightsquigarrow ba$ (τ_2) are applied on top of the sentence $abcd$ (s). As it can be seen, the resulting word lattice contains the path of the original sentence $s : abcd$, as well as the additional paths appeared by the composition of reordering rules: $\tau_1(s) : cab$, $\tau_2(s) : ba$ and $\tau_2(\tau_1(s)) : cba$.

Part-of-speech (POS) and syntactic information are used to increase the generalization power of our rules. Hence, instead of raw words, the LHS of the reordering rules typically make reference to POS-tags patterns, or to dependency sub-trees.

For instance, the rule $NN\ JJ \rightsquigarrow JJ\ NN$ is defined in terms of POS-tags, and produces the swap of the sequence *noun adjective* that is observed for the pair French-to-English. Additional details regarding the syntax-based rules are given in section 3.

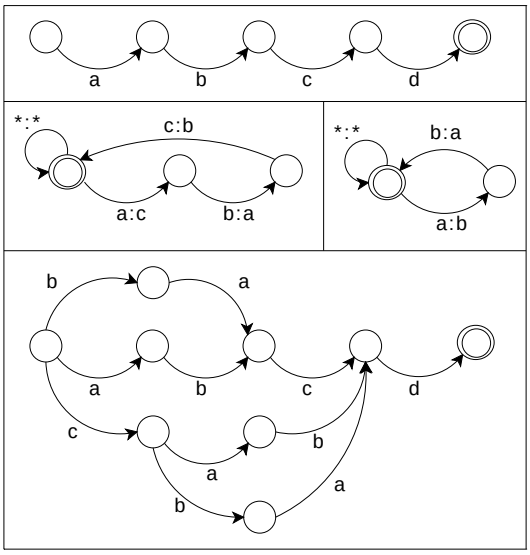


Figure 1: *Initial linear automaton (top). Reordering rules in the form of string transducers (middle) and final word lattice after rule composition.*

For the experiments reported in this paper, we consider that all paths in the input lattice are equally likely, a simplification we may wish to remove in further research.

3 Translation units with gaps

In this section we give details of the gappy translation units introduced in this work.

3.1 Split rules and reordering

Some phrase-based systems have been able to introduce some levels of syntactical information. In (Habash, 2007) the author employs automatically learned syntactic reordering rules to preprocess the input, aiming at solving the reordering problem, before passing the reordered input to a phrase-based decoder for Arabic-English translation. However, this kind of systems cannot produce the translation

needed in our original English-to-French example because of the left-to-right decoding approach used in the underlying system. Translation is sequentially composed from left to right, and none of the word orderings of the source sentence, *don't + want* and *want + don't*, produces the desired translation. Instead, they produce respectively: *ne pas + veux* and *veux + ne pas*.

We propose a method that allows phrase-based systems to introduce gappy units similar to those typically employed in hierarchical systems, while keeping the left-to-right decoding approach.

To collect gappy units, we analyze the (symmetric) word alignments of the training corpus. The method basically consists of identifying, in the source sentence, single tokens translated into multiple ($n > 1$) non-contiguous target tokens. Figure 2 shows an example.

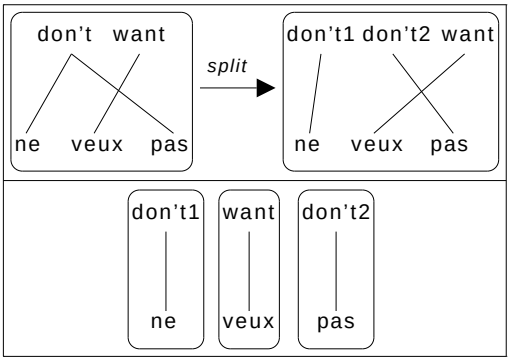


Figure 2: *Original tuple (top left), introduction of split words (top right) and tuples obtained after reordering source words (bottom).*

In the example, the English token *don't* is translated into a sequence of discontinuous word segments *ne ... pas*. Once identified, the original source token is split so as to match the number of discontinuous segments. To continue with our example, *don't* is split into $don't^1$ and $don't^2$ to match the two discontinuous segments *ne...pas*. Hence, similar to (Crego and Mariño, 2007b), we aim at monotonizing the word-to-word alignment, the main novelty being here the introduction of split tokens.

As it can be seen in the example, the target side of translation units remains unchanged, meaning that we can continue to generate the target in left-to-right fashion. Word reorderings and split words are introduced in the source sentence only, motivating the use of a word lattice. During training, the alignment is entirely monotonized before extracting tuples, only keeping those *one-to-many* and *many-to-one* alignments where the tokens on the *many* are contiguous; when this is not the case, splitting takes place.

Note that when translating the same example in the opposite direction, that is from French to English, the right translation is achieved without needing to split tokens. In such a case, the system would proceed by first reordering source words, obtaining *ne pas veux*, and then monotonically translating using the units: *ne pas : don't* and *veux : want*, yielding the right translation *don't want*.

When decoding test sentences, the word lattice is used to encode the most promising reorderings/splits of the input sentence, so as to reproduce the modifications introduced in the source sentences of the training corpus (as shown in figure 2). Thus, we slightly extend the reordering formalism introduced in 2 to allow the insertion of split tokens. Following the previous example, the new rule consists of:

$$don't \;\; want \;\; \rightsquigarrow \;\; don't^1 \;\; want \;\; don't^2$$

meaning that whenever you find in the input sentence the word sequence *don't want*, the input lattice is extended with the path $don't^1$ *want* $don't^2$, as represented on figure 3.

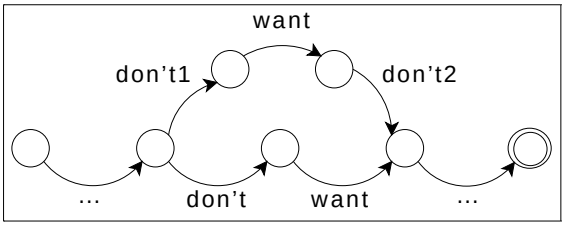


Figure 3: *Monotonic input graph extended with a split rule.*

So far, the method presented does not produce gappy units, but standard tuples with higher monotonization levels. However, with the addition of split rules, they become very similar to the units used in hierarchical translation systems. Note that the resulting extended input graph (figure 3) contains exactly

the units extracted by the splitting procedure (figure 2 bottom).

The fully lexicalized split rules previously introduced would however be useless, failing to generalize to novel patterns. Therefore, as is done with "standard" reordering rules, split rules are defined over patterns of POS tags, instead words. Of course, the identity of split word has to be preserved, as it would make no sense to split, during decoding, words for which no translation units have been collected in training. Finally, the split rule induced for the previous example is:

$$don't \;\; V \;\; \rightsquigarrow \;\; don't^1 \;\; V \;\; don't^2$$

where V is a POS tag standing for a verb.

This strategy has two additional benefits. First, it yields smaller translation units, whose probability are better estimated. Going back to the example of figure 2, the original translation unit (left) is larger than the new one (right), and more likely to cause estimation problems. Second, it allows to better use the information available in the training corpus. To see why, consider again our running example. Leaving the original unit undecomposed prevents to extract the match between *want* and *veux*, which is correctly extracted in the novel formalism.

In the next section, we detail how the generalization power of split/reordering rules can be further increased by using dependency parse trees.

3.2 Syntax aware split rules

Syntactic reordering rules employed in this work are similar to those detailed in (Crego and Mariño, 2007b). These rules introduce reorderings at the level of syntactic nodes. Hence, long reorderings can be achieved with short rules, as nodes may dominate arbitrary long sequences of words. Thus, the LHS of the rules is referred to the parse nodes of the original source sentences, while the RHS specifies the permutation that is introduced. Figure 4 shows the parse tree and POS tags of the Chinese sentence: *Aozhou shi yu Beihan you bangjiao de shaoshu guojia zhiyi*, an example borrowed from (Chiang, 2007).

Figure 5 illustrates how, by applying three rules to the previous Chinese example, we can get the reorderings/split required to derive the correct English translation: *Australia is one of the few countries that have diplomatic relations with North Korea*. As previously stated, rules (FSTs) are sorted before applied (composed). Note that in the case of syntactic rules, the length of a rule is based on the number of words appearing in the LHS of the rule.

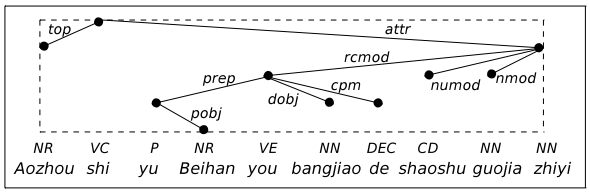


Figure 4: *Dependency parse tree and POS tags of the Chinese sentence: 'Aozhou shi yu Beihan you bangjiao de shaoshu guojia zhiyi'.*

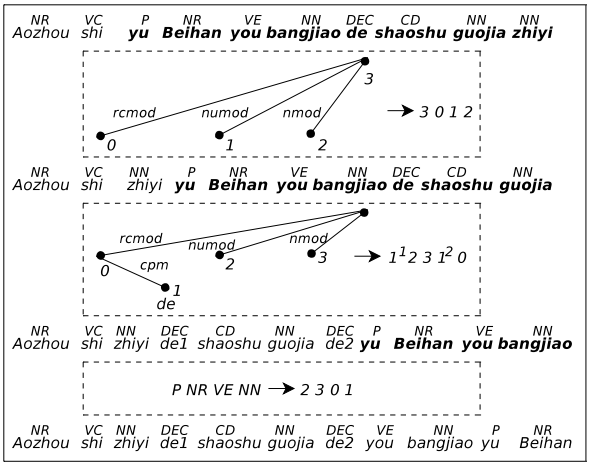


Figure 5: *Chinese sentence rewritten by means of reordering/split rules.*

Considering the first rule applied in figure 5, the tree in its LHS contains four nodes (eight words), which cover the following sequences of Chinese words: *yu Beihan you bangjiao de*, *shaoshu*, *guojia* and *zhiyi*. Words matched by the rules are displayed above the rules using bold characters.

Note that equivalently to POS rules, words to be split in syntactical rules appear fully lexicalized. The second rule in figure 5 splits the word *de*. Thus, it appears fully lexicalized in the LHS of the rule.

Finally, the last rule is formed of POS tags. It reorders the words *yu Beihan you bangjiao* into *you bangjiao yu Beihan*. The monotonic translation of the resulting reordered path yields the correct English translation.

Syntactical reordering/split rules are automatically extracted from the training bi-texts, making use of the word-to-word alignments and the source dependency trees.

To conclude this section, notice that gappy units introduced in this work are only those that are motivated by word structures where words of the source side are aligned to multiple non-contiguous words of the target side. As a result, we approximate the behavior of a hierarchical system employing only a very limited set of rule patterns.

4 Experiments

In this section, we give details regarding the evaluation framework and report on the experimental work carried out to evaluate the improvements.

4.1 Evaluation Framework

We have used the BTEC (Takezawa et al., 2002) corpus focusing on translations from Chinese to English. It consists of the data made available for the IWSLT 2007 evaluation campaign. Some statistics regarding the corpora used, namely number of sentences, words, vocabulary, average sentence length and number of references per language are shown in table 1.

	Sent	Words	Voc	Avg	Refs
Train					
en	40k	377k	11k	9.5	1
zh		354k	9,6k	8.9	
Tune / Test (zh)					
tune	506	3,564	871	7	16
tst2	500	3,608	921	7.22	16
tst3	506	3,889	916	7.69	16
tst4	489	5,476	1,094	11.2	7
tst5	500	5,846	1,292	11.69	7
tst6	489	3,325	864	6.8	6

Table 1: *BTEC Corpus (Chinese-to-English).*

Chinese words were segmented by means of the ICTCLAS (Zhang et al., 2003) tagger/segmenter. Word alignments were computed for the training data in the original word order, using GIZA++[3]. The grow-final-diag-and heuristic is used to refine the alignments before the translation units extraction. The Chinese side was parsed using the freely available Stanford Chinese Dependency Parser[4]. We have used the SRILM toolkit[5] to estimate the N-gram language models, using respectively 4 and 5 as n-gram orders for the translation LM and target LM (Kneser-Ney smoothing and interpolation of lower and higher n-grams are always used).

For tuning, optimal log-linear coefficients were found using an in-house implementation of the downhill SIMPLEX method. The BLEU score was used as the objective function.

4.2 Results

Accuracy results are reported for different configurations in table 2. System configurations consist of: **base** for which translation units do not introduce the ability to split source words into multiple tokens, and **+split** where the previous technique is used. The **POS** configuration employs POS tags in the source side of the reordering rules while **+SYN** employs both POS tag and syntactic rules.

Set	base		+split		Moses
	POS	+SYN	POS	+SYN	
tst2	47.25	48.15	47.42	**48.39**	48.14
tst3	55.82	56.88	56.44	**57.17**	55.95
tst4	15.72	16.82	16.48	17.08	**18.06**
tst5	15.89	16.32	16.34	**16.89**	15.91
tst6	29.56	30.81	29.81	31.67	**31.76**

Table 2: Accuracy results measured using the BLEU score.

The last column shows accuracy results obtained by **Moses** (Koehn et al., 2007), a state-of-the-art phrase-based SMT system.

It is worth saying that the Moses system was built using the same data sets and alignments that were used for our system (Moses performs lexicalized reordering with a maximum reordering distance of 8 words). In this case, we run a different optimization for each of the system configurations. BLEU confidence intervals range depending on the test set approximately from ±2.0 to ±3.0 points BLEU.

As it can be seen, the system built using the **+split** technique obtains higher accuracy results than the baseline one (**base**), in all test

[3] www.fjoch.com/GIZA++

[4] nlp.stanford.edu/downloads/lex-parser.shtml

[5] www.speech.sri.com/projects/srilm

sets and for both reordering rule configurations (**POS** and **+SYN**).

Even if results show a clear tendency to highly score the **+split** system, differences in all BLEU results fall within the confidence margin. However, when inspecting translations obtained by the system **+split +SYN**, we find several examples, such as the one shown in figure 6, where the decoder succeeds to apply the proposed gappy units.

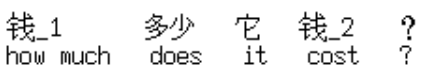


Figure 6: *Sequence of translation units output by the decoder.*

As it can be seen, motivated by a gappy unit, the first Chinese word is translated in two distant steps, yielding *how much* and *cost* respectively. The gap between both fragments is correctly filled by the English words *does it* as translation of the second and third Chinese words.

Considering the **base** systems, the same translation could only be produced if the first three Chinese words had been seen in training aligned to *how much does it*. In other words, larger units are needed to account for the correct translation.

The increment in the total number of translation units extracted when moving from the **base** to the **+split** configurations (from $267k$ to $285k$), as well as the increment in units used to translate the test sets (from $18,345$ to $19,150$) supports the fact that higher monotonizations levels of the training corpus have been achieved. All together, the resulting vocabulary of translation units, including all the new split units ($13,706$), contains $63,036$ units to be compared with the $56,046$ units in the baseline system.

Considering search efficiency, decoding time was increased about 1.5 times when building the system using the **split** technique, for both reordering rule configurations (**POS** and **+SYN**). Using gappy translation units does not increase the complexity of the search.

5 Conclusions and Further Work

In this paper, we have presented an extension to a bilingual n-gram translation system in which we allow translation units with gaps. The use of word lattices allowed us to introduce the concept of gappy translation units into an n-gram-based system, as an attempt to bridge the gap between phrase-based and hierarchical systems. Our decoder additionally benefits from the simplicity of left-to-right decoders, in contrast to the cost in complexity incurred by performing decoding as parsing. This have been achieved by means of standard tuples tightly coupled with reordering/split rules, introduced into the overall search through an input word lattice.

Our small but consistent accuracy improvements can mainly be attributed to the fact that a higher level of monotonization of the training corpus allows the extraction of smaller/more reusable units. As explained above, the split/reordering rules used in this study are costless, meaning that all reorderings are equally likely. As a consequence, the reward of using a split rule only comes from the translation models' score, which are computed separately for each instance of a split token. We believe that devising an appropriate weighting scheme for these split/reordering rules is needed to take full advantage of the extra expressiveness allowed by gappy units.

With the objective that our translation model highly benefits from the advantages of additional context, each gappy translation unit must be entirely weighted with a single probability. Instead, in our current implementation, each gappy unit is multiply weighted with partial probabilities. An open issue to definitely tackle in further research.

Additionally, we believe that the slight improvements achieved can be increased if additional gappy units are acquired from bilingual structures other than the one-to-many employed in the present experiments. We plan to extend the framework proposed in this paper with more complex gappy units, similiar to those used by hierarchical MT systems, thereby, taking full advantage of additional translation context provided by these units. We also plan to further investigate other aspects of hierarchical units, such as different

levels of lexicalization in both the source and the target side.

Acknowledgments

This work has been partially funded by OSEO under the Quaero program.

References

Brown, P., S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Crego, J.M. and J.B. Mariño. 2007a. Extending marie: an n-gram-based smt decoder. *45rd Annual Meeting of the Association for Computational Linguistics*, April.

Crego, J.M. and J.B. Mariño. 2007b. Syntax-enhanced n-gram-based smt. *Proc. of the MT Summit XI*, pages 111–118, September.

Habash, N. 2007. Syntactic preprocessing for statistical machine translation. *Proc. of the MT Summit XI*, September.

Knight, Kevin. 2008. Capturing practical natural language transformations. *Machine Translation*, 21(2):121–133.

Koehn, Ph., F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. *Proc. of the Human Language Technology Conference, HLT-NAACL'2003*, May.

Koehn, Philipp, 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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Mariño, J.B., R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, and M.R. Costajussà. 2006. N-gram based machine translation. *Computational Linguistics*, 32(4):527–549.

Melamed, D. 2004. Statistical machine translation by parsing. *42nd Annual Meeting of the Association for Computational Linguistics*, pages 653–661, July.

Och, F.J., Ch. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. *Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, June.

Simard, M., N. Cancedda, B. Cavestro, M. Dymetman, E. Gaussier, C. Goutte, K. Yamada, P. Langlais, and A. Mauser. 2005. Translating with non-contiguous phrases. pages 755 – 762, October 6-8.

Takezawa, T., E. Sumita, F. Sugaya, H Yamamoto, and S. Yamamoto. 2002. Toward a broad-coverage bilingual curpus for speech translation of travel conversations in the real world. *3rd Int. Conf. on Language Resources and Evaluation, LREC'02*, pages 147–152, May.

Venugopal, Ashish, Andreas Zollmann, and Vogel Stephan. 2007. An efficient two-pass approach to synchronous-CFG driven statistical MT. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 500–507, Rochester, New York, April. Association for Computational Linguistics.

Watanabe, T., H. Tsukada, and H Isozaki. 2006. Left-to-right target generation for hierarchical phrase-based translation. *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, July.

Zens, R., F.J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In Jarke, M., J. Koehler, and G. Lakemeyer, editors, *KI - 2002: Advances in artificial intelligence*, volume LNAI 2479, pages 18–32. Springer Verlag, September.

Zhang, H., H. Yu, D. Xiong, and Q. Liu. 2003. HHMM-based chinese lexical analyzer ictclas. In *Proc. of the 2nd SIGHAN Workshop on Chinese language processing*, pages 184–187, Sapporo, Japan.

Zollmann, Andreas and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June. Association for Computational Linguistics.

Zollmann, Andreas, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1145–1152, Manchester, UK, August. Coling 2008 Organizing Committee.

Relevance of Different Segmentation Options on Spanish-Basque SMT

Arantza Díaz de Ilarraza, Gorka Labaka and Kepa Sarasola
Euskal Herriko Univertsitatea/Universidad del País Vasco
jipdisaa@ehu.es, gorka.labaka@ehu.es, jipsagak@ehu.es

Abstract

Segmentation is widely used in adapting Statistical Machine Translation to highly inflected languages as Basque. The way this segmentation is carried out impacts on the quality of the translation. In order to look for the most adequate segmentation for a Spanish-Basque system, we have tried different segmentation options and analyzed their effects on the translation quality.

Although all segmentation options used in this work are based on the same morphological analysis, translation quality varies significantly depending on the segmentation criteria used. Most of the segmentation options outperform the baseline according to all metrics, except the one which splits words according the morpheme boundaries. From here we can conclude the importance of the development of the segmentation criteria in SMT.

1 Introduction

In this paper we present the work done for adapting a baseline SMT system to carry out the translation into a morphologically-rich agglutinative language such as Basque. In translation from Spanish to Basque, some Spanish words, such as prepositions or articles, correspond to Basque suffixes, and, in case of ellipsis, more than one of those suffixes can be added to the same word. In this way, based on the Basque lemma 'etxe' */house/* we can generate 'etxeko' */of the house/*, 'etxekoa' */the one of the house/*, 'etxekoarengana' */towards the one of the house/* and so on.

Besides, Basque is a low-density language and there are few corpora available comparing to other languages more widely used as Spanish, English, or Chinese. For instance, the parallel corpus available for this work is 1M word for Basque (1.2M words for Spanish), much smaller than the corpora usually used on public evaluation campaigns such as NIST.

In order to deal with the problems presented above, we have split up Basque words into the lemma and some tags which represent the morphological information expressed on the inflection. Dividing Basque words in this way, we expect to reduce the sparseness produced by the agglutinative being of Basque and the small amount of training data.

Anyway, there are several options to define Basque segmentation. For example, considering all the suffixes all together as a unique segment, considering each suffix as a different segment, or considering any other of their intermediate combinations. In order to define the most adequate segmentation for our Spanish-Basque system, we have tried some of those segmentation options and have measured their impact on the translation quality.

The remainder of this paper is organized as follows. In Section 2, we present a brief analysis of previous works adapting SMT to highly inflected languages. In Section 3, we describe the systems developed for this paper (the baseline and the morpheme based systems) and the different segmentation used by those systems. In Section 4, we evaluate the different systems, and report and discuss our experimental results. Section 5 concludes the paper and gives avenues for future work.

Proceedings of the 13th Annual Conference of the EAMT, pages 74–80,
Barcelona, May 2009

2 Related work

Many researchers have tried to use morphological information in improving machine translation quality. In (Koehn and Knight, 2003), the authors got improvements splitting compounds in German. Nießen and Ney (2004) achieved a similar level of alignment quality with a smaller corpora restructuring the source based on morpho-syntactic information when translating from German to English. More recently, on (Goldwater and McClosky, 2005) the authors achieved improvements in Czech-English MT optimizing a set of possible source transformations, incorporating morphology.

In general most experiments are focused on translating from morphologically rich languages into English. But last years some works have experimented on the opposite direction. For example, in (Ramanathan et al., 2008), the authors segmented Hindi in English-Hindi statistical machine translation separating suffixes and lemmas and, in combination with the reordering of the source words based on English syntactic analysis, they got a significant improvement both in automatic and human evaluation metrics. In a similar way Oflazer and El-Kahlout (2007) also segmented Turkish words when translate from English. The isolated use of segmentation does not get any improvement at translation, but combining segmentation with a word-level language model (incorporated by using n-best list re-scoring) and setting as unlimited the value of the *distortion limit* (in order to deal with the great order difference between both languages) they achieve a significant improvement over the baseline.

Segmentation is the most usual way to translate into highly inflected languages, but other approaches have been also tried. In (Bojar, 2007) factored translation have been used on English-Czech translation. Words of both languages are tagged with morphological information creating different factors which are translated independently and combined in a generation stage. Finally, in (Minkov et al., 2007) the authors have divided translation in two steps where they first use usual SMT system to translate from English to Russian lemmas and in a second step they decide the inflection of each lemma using bilingual information.

3 SMT systems

The main deal of this work is to measure the impact of different segmentation options on a Spanish-Basque SMT system. In order to measure this impact we have compared the quality of the baseline system which does not use segmentation at all, with systems that use different segmentation options. the development of those systems has been carried out using freely available tools:

- GIZA++ toolkit (Och and H. Ney, 2003) was used for training the word alignment.
- SRILM toolkit (Stolcke, 2002) was used for building the language model.
- Moses Decoder (Koehn et al., 2007) was used for translating the test sentences.

3.1 Baseline

We have trained Moses on the tokenized corpus (without any segmentation) as baseline system. Moses and the scripts provided with it allow to easily train a state-of-the-art phrase-based SMT system. We have used a log-linear (Och and Ney, 2002) combination of several common feature functions: phrase translation probabilities (in both directions), word-based translation probabilities (lexicon model, in both directions), a phrase length penalty and a target language model.

The decoder also relies on a target language model. The language model is a simple 5-gram language model trained on the Basque portion of the training data, using the SRI Language Modeling Toolkit, with modified Kneser-Ney smoothing. Finally, we have also used a lexical reordering model (one of the advanced features provided by Moses[1]), trained using Moses scripts and *'msd-bidirectional-fe'* option. The general design of the baseline system is presented on Figure 1.

Moses also implements Minimum-Error-Rate Training (Och, 2003) within a log-linear framework for parameter optimization. The metric used to carry out this optimization is BLEU (Papineni et al., 2002).

3.2 Morpheme-based statistical machine translation

Basque is an agglutinative language, so words may be made up several morphemes. Those morphemes are added as suffixes to the last word of

[1] http://www.statmt.org/moses/?n=Moses.AdvancedFeatures

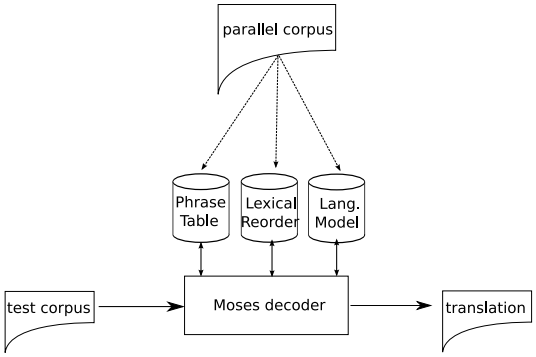


Figure 1: Basic design of a SMT system

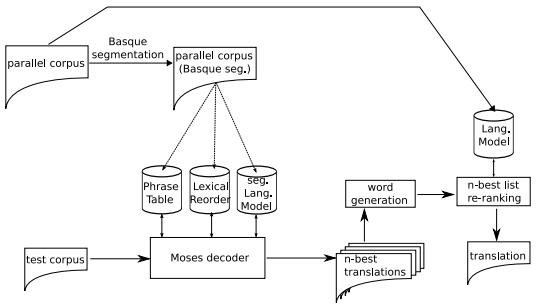


Figure 2: Design of the morpheme-based SMT system

noun phrases and verbal chains. Suffixes represent the morpho-syntactic information associated to the phrase, such as number, definiteness, grammar case and postposition.

As a consequence, many words only occur once in the training corpus, leading to serious sparseness problems when extracting statistics from the data. In order to overcome this problem, we segmented each word into a sequence of morphemes, and then we worked at this representation level. Working at the morpheme level we reduced the number of tokens that occur only once and, at the same time, we reduce the 1-to-n alignments. Although 1-to-n alignments are allowed in IBM model 4, training can be harmed when the parallel corpus contains many cases.

Adapting the baseline system to work at the morpheme level mainly consists on training Moses on the segmented text (same training options are used in baseline and morpheme-based systems). The system trained on these data will generate a sequence of morphemes as output and a generation post-process will be necessary in order to obtain the final Basque text. After generation, we have integrated a word-level language model using n-best list re-ranking. The general design of the morpheme-based system is presented on Figure 2.

3.2.1 Segmentation options for Basque

Segmentation of Basque words can be made in different ways and we want to measure the impact those segmentation options have on the translation quality. In order to measure this impact, we have tried different ways to segment Basque words and we have trained a different morpheme-based system on each segmentation.

The different segmentation options we have tried are all based on the analysis obtained by Eustagger (Aduriz and Díaz de Ilarraza, 2003), a tagger for Basque based on two-level morphology (Koskeniemmi, 1983) and statistical disambiguation. Based on those analysis we have divided each Basque word in different ways. From the most fine-grained segmentation, where each morpheme is represented as a token, to the most coarse-grained segmentation where all morphemes linked to the same lemma are put together in an unique token. Figure3 shows an analysis obtained by Eustagger the lemma and the morphological information added by the morphemes is represented marking the morphemes boundaries with a '+'.

Following we define the four segmentation options we are experimenting with.

Eustagger Segmentation: In our first approach we have strictly based on the lexicon of Eustagger, and we have created a separate token for each morpheme recognized by the analyzer. This lexicon has been created following a linguistic perspective and, although it has been proved very useful for the develop of several applications, it is probably not the most adequate for this work. As the lexicon is very fine-grained, some suffixes, which could be considered as a unique morpheme, are represented as a concatenation of several fine-grained morphemes in the Eustagger lexicon. Furthermore, some of those morphemes have not any effect on the word form, and they only adds some morphological features. Figure 3 shows segmentation of 'aukeratzerakoan' */at the election time/* word according to the segmentation produced by Eustagger.

One suffix per word: Taking into account that the Eustagger lexicon is too fine-grained and that it generates too many tokens at segmentation, our next approach consisted on putting together all suffixes linked to a lemma in one token. So, at splitting one Basque word we will generate at most

Analysis	aukeratu<adi><sin>+<adize>+<ala><gel>+<ine>				
Eustagger seg.	aukeratu<adi><sin>	+<adize>	+<ala>	+<gel>	+<ine>
Automatic seg.	aukeratu<adi><sin>	+<adize><ala>	+<gel>	+<ine>	
Hand defined seg.	aukeratu<adi><sin><adize>	+<ala><gel><ine>			
OneSuffix seg.	aukeratu<adi><sin>	+<adize><ala><gel><ine>			

Figure 3: Analysis obtained by Eustagger for 'aukeratzerakoan' */at the election time/* word. And the distinct segmentation inferred from it.

three tokens (prefixes, lemma and suffixes). We can see 'aukeratzerakoan' */at the election time/* word's segmentation on Figure 3.

Manual morpheme-grouping: After realizing the impact of the segmentation in translation, we tried to obtain an intermediate segmentation which optimizes the translation quality. Our first attempt consists on defining by hand which morphemes can be grouped together in one token and which ones can be considered a token by their own. In order to decide which morphemes to group, we have analyzed the alignment errors occurred at previous segmentation experiments, defining a small amount of rules to grouping morphemes. For instance, '+<adize>'[2] morpheme is usually wrongly aligned when it is considered as a token, so we have decided to join it to the lemma at segmentation. On Figure 3 we can see the segmentation corresponding to 'aukeratzerakoan' */at the election time/* word.

Automatic morpheme-grouping: Anyway, the morpheme-grouping defined by hand depends on the language pair and if we change it, we should redefine the grouping criteria, analyzing again the detected errors. So, in order to find a language independent way to define the most appropriate segmentation, we focus our research in establishing a statistical method to decide which morphemes have to be put into the same token. We observed that the morphemes which generates most of the errors are those which have not their own *meaning*, those that *need* another morpheme to complete their meaning. We thought on using the *mutual information* metric in order to measure statistical dependence between two morphemes. We will group those morphemes that are more dependent than a threshold. On this experiment we tried different thresholds and we obtained the best results when it is set to 0.5 (value that involve grouping most of the morphemes). In Figure 3 we can see 'aukeratzerakoan' */at the election time/* word segmented in this way.

[2]suffix for verb normalisation

3.2.2 Generating words from morphemes

When working at the morpheme level, the output of our SMT system is a sequence of morphemes. In order to produce the proper Basque text, we need to generate the words based on this sequence, so the output of the SMT system is post-processed to produce the final Basque translation.

To develop generation post-processing, we reuse the lexicon and two-level rules of our morphological tool Eustagger. The same generation engine is useful for all the segmentation options defined in section 3.2.1 since we have produced them based on the same analysis. However, we have to face two main problems:

- Unknown lemmas: some lemmas such as proper names are not in the Eustagger lexicon and could not be generated by it. To solve this problem and to be able to generate inflection of those words, the synthesis component has been enriched with default rules for unknown lemmas.

- Invalid sequences of morphemes: the output of the SMT system is not necessarily a well-formed sequence from a morphological point of view. For example, morphemes can be generated in a wrong order or they can be missed or misplaced (i.e. a nominal inflection can be assigned to a verb). In the current work, we did not try to correct these mistakes, and when the generation module can not generate a word it outputs the lemma without any inflection. A more refined treatment is left for future work.

3.3 Incorporation of word-level language model

When training our SMT system over the segmented test the language model used in decoding is a language model of morphemes (or groups of morphemes depending on the segmentation option). Real words are not available at decoding, but, after generation we can incorporate a second

		sentences	words	morph	word-vocabulary	morph-vocabulary
training	Spanish Basque	58,202	1,284,089 1,010,545	- 1,699,988	46,636 87,763	- 35,316
development	Spanish Basque	1,456	32,740 25,778	- 43,434	7,074 9,030	- 5,367
test	Spanish Basque	1,446	31,002 24,372	- 41,080	6,838 8,695	- 5,170

Table 1: Some statistics of the corpora.

language model based on words. The most appropriate way to incorporate the word-level language model is using n-best list as was done in (Oflazer and El-Kahlout, 2007). We ask Moses to produce a n-best list, and after generating the final translation based on Moses output, we estimate the new cost of each translation incorporating word-level language model. Once new cost is calculated the sentence with the lowest cost is selected as the final translation.

The weight for the word-level language model is optimized at Minimum Error Rate Training with the weights of the rest of the models. Minimum Error Rate Training procedure has been modified to post-process Moses output and to include word-level language model weight at optimization process.

4 Experimental results

4.1 Data and evaluation

In order to carry out this experiment we used the *Consumer Eroski* parallel corpus. This corpus is a collection of 1036 articles written in Spanish (January 1998 to May 2005, Consumer Eroski magazine, http://revista.consumer.es) along with their Basque, Catalan and Galician translations. It contains more than 1,200,000 Spanish words and more than 1,000,000 Basque words. This corpus was automatically aligned at sentence level[3] and it is available[4] for research. Consumer Eroski magazine is composed by the articles which compare the quality and prices of commercial products and brands.

We have divided this corpus in three sets, training set (60,000 sentences), development set (1,500 sentences) and test set (1,500 sentences), more detailed statistics on Table 1.

[3]corpus was collected and aligned by Asier Alcázar from the University of Missouri-Columbia

[4]The Consumer corpus is accessible on-line via Universidade de Vigo (http://sli.uvigo.es/CLUVI/, public access) and Universidad de Deusto (http://www.deli.deusto.es, research intranet).

In order to assess the quality of the translation obtained using the systems, we used four automatic evaluation metrics. We report two accuracy measures: BLEU, and NIST (Doddington, 2002); and two error measures: Word Error Rate (WER) and Position independent word Error Rate (PER). In our test set, we have access to one Basque reference translation per sentence. Evaluation is performed in a case-insensitive manner.

4.2 Results

The evaluation results for the test corpus is reported in Table 2. These results show that the differences at segmentation have a significant impact at translation quality. Segmenting words according to the morphemes boundaries of the Eustagger lexicon does not involve any improvement. Compared to the baseline, which did not use any segmentation, the results obtained for the evaluation metrics are not consistent and varies depending on the metric. According to BLEU segmentation harms translation, but according the rest of the metrics the segmentation slightly improves translation, but this improvement is probably not statistically significant.

The rest of the segmentation options, which are based on the same analysis of Eustagger and contains the same morpheme sequences, consistently outperforms baseline according to all the metrics. Best results are obtained using the hand defined criteria (based on the alignment errors), but automatically defined segmentation criteria obtains similar results.

Due to the small differences on the results obtained for the evaluation metrics we have carried out a statistical significance test (Zhang et al., May 2004) over BLEU. According with this, the system using hand defined segmentation significantly outperforms both the system using OneSuffix segmentation and the system using segmentation based on mutual information. Difference between the system using OneSuffix segmentation and the system based on mutual information are

	BLEU	NIST	WER	PER
Baseline	10.78	4.52	80.46	61.34
MorphemeBased-Eustagger	10.52	4.55	79.18	61.03
MorphemeBased-OneSuffix	11.24	**4.74**	**78.07**	**59.35**
MorphemeBased-AutoGrouping	11.24	4.66	79.15	60.42
MorphemeBased-HandGrouping	**11.36**	4.69	78.92	60.23

Table 2: BLEU, NIST, WER and PER evaluation metrics.

Segmentation option	Running tokens	Vocabulary size	BLEU
No Segmentation	1,010,545	87,763	10.78
Hand Defined grouping	1,546,304	40,288	**11.36**
One Suffix per word	1,558,927	36,122	11.24
Statistical morph. grouping	1,580,551	35,549	11.24
Eustagger morph. boundaries	1,699,988	35,316	10.52

Table 3: Correlation between token amount on the train corpus and BLEU evaluation results

not statistically significant.

Finally, given the low scores obtained, we would like to make two additional remarks. First, it shows the difficulty of the task of translating into Basque, which is due to the strong syntactic differences with Spanish. Second, the evaluation based on words (or n-grams of words) always gives lower scores to agglutinative languages like Basque. Often one Basque word is equivalent to two or three Spanish or English words, so a 3-gram matching in Basque is harder to obtain having a highly negative effect on the automatic evaluation metrics.

4.3 Correlation between segmentation and BLEU

Analyzing the obtained results, we have realized that there are a correlation between the amount of tokens generated at segmentation and the results obtained at evaluation. Before segmentation, there are 1M words for Basque, which together with the 1.2M words for Spanish, make the word alignment more difficult (due to the 1-to-n alignment amount). Anyway, after segmenting the Basque words according with the morpheme boundaries of Eustagger, the Basque text contains 1.7M tokens (the same alignment problem is generated but in the opposite direction) see Table 3.

Intermediate segmentation options, where morphemes marked by Eustagger are grouped in different ways, get better results when the amount of the generated tokens is closer to the amount of tokens we have in Spanish part. We leave for future work to experiment ways to reduce the different number of tokens of both languages.

5 Conclusions and Future work

We have proved that the quality of the translation varies significantly when applying different options for word segmentation. Based on the same output of morphological analyzer, we have segmented words in different ways creating more fine or coarse grained segments (from one token per each morpheme to a unique token for all suffixes of a word). Surprisingly, the criteria based on considering each morpheme as a separate token obtains worse results than the system without segmentation. Other segmentation options outperforms the baseline, getting the best results with a hand defined intermediate grouping based on an alignment error analysis.

Anyway, the work done by hand is language dependent and could not be reused for a different pair of languages, so we also tried a statistical way to determine the morpheme grouping criteria which gets almost as accurate results as those obtained with the hand defined criterion. So we could use this statistical grouping criteria to adapt our system to a different language pair such as English-Basque.

As future work, we thought on trying a different measure to determine the statistical independence of the morphemes, as χ^2. Besides, as the dependence between morphemes is calculated on the monolingual text, a bigger monolingual corpus could be used (instead of using just the Basque side of the bilingual corpus) for this.

Taking into account the obtained correlation between the token amount and translation quality. We want to redefine the segmentation criteria to reduce the amount of tokens obtained. In such a way that the difference in the number of tokens of

both languages would be reduced.

Acknowledgement

This research was supported in part by the Spanish Ministry of Education and Science (OpenMT: Open Source Machine Translation using hybrid methods, TIN2006-15307-C03-01) and the Regional Branch of the Basque Government (AnHITZ 2006: Language Technologies for Multilingual Interaction in Intelligent Environments., IE06-185). Gorka Labaka is supported by a PhD grant from the Basque Government (grant code, BFI05.326).

Consumer corpus has been kindly supplied by Asier Alcázar from the University of Missouri-Columbia and by Eroski Fundazioa.

References

Aduriz, I. and A. Díaz de Ilarraza. 2003. Morphosyntactic disambiguation ands shallow parsing in Computational Processing of Basque. In *Inquiries into the lexicon-syntax relations in Basque. Bernarrd Oyharabal (Ed.)*, Bilbao.

Bojar, Ondrej. 2007. English-to-Czech Factored Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic.

Doddington, G. 2002. Automatic evaluation of Machine Translation quality using n-gram cooccurrence statistics. In *Proceedings of HLT 2002*, San Diego, CA.

Goldwater, S. and D. McClosky. 2005. Improving Statistical MT Through Morphological Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Vancouver.

Koehn, P. and K. Knight. 2003. Empirical Methods for compound splitting. In *Proceedings of EACL 2003*, Budapest, Hungary.

Koehn, Philipp, 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)*, Prague, Czech Republic, June.

Koskeniemmi, K. 1983. Two-level Model for Morphological Analysis. In *Proceedings of the Eigth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany.

Minkov, E., K. Toutanova, and H. Suzuki. 2007. Generating Complex Mophology for Machin Translation. In *Proceedings of 45th ACL*, Prague, Czech Republic.

Nießen, S. and H. Ney. 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Comput. Linguist.*, 30(2):181–204.

Och, F. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, Franz Josef and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *ACL*, pages 295–302.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

Oflazer, Kemal and Ilknur Durgar El-Kahlout. 2007. Exploring Different Representation Units in English-to-Turkish Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic.

Papineni, K., S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th ACL*, Philadelphia, PA.

Ramanathan, Ananthakrishnan, Pushpak Bhattacharyya, Jayprasad Hegde, Ritesh M.Shah, and Sasikumar M. 2008. Simple Syntactic and Mophological Processing Can Help English-Hindi Statistical Machine Translation. In *IJCNLP 2008: Third International Joint Conference on Natural Language Processing*, Hyderabad, India.

Stolcke, Andreas. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September.

Zhang, Ying, Stephan Vogel, and Alex Waibel. May 2004. Interpreting Bleu/NIST scores: How much improvement do we need to have a better system? In *Proceedings of LREC 2004*, Lisbon, Portugal.

English-Latvian Toponym Processing:

Translation Strategies and Linguistic Patterns

Tatiana Gornostay
Tilde, Latvia
tatjana.gornostaja@tilde.lv

Inguna Skadiņa
Tilde, Latvia
inguna.skadina@tilde.lv

Abstract

The paper presents a study of a challenging task in machine translation and cross-language information retrieval – translation of toponyms. Due to their linguistic and extra-linguistic nature, toponyms deserve a special treatment. The overall translation process includes two stages of processing: dictionary-based and out-of-vocabulary toponym translation. The latter is divided into three steps: source string normalisation, translation, and target string normalisation. The translation process implies an application of translation strategies and linguistic toponym translation patterns. Possible translation strategies, including transliteration and translation *per se* along with combined strategies, and linguistic toponym translation patterns, including multi-word patterns as well, were investigated and implemented for English-Latvian machine translation. 10,000 The UK-related toponyms from Geonames were selected for a development set. The evaluation of output quality on basis of a test set has showed 67% accuracy in out-of-vocabulary translation: 58% on a set containing one-word toponymic units and 81% on a multi-word test set.

1 Introduction

The paper presents a study of a challenging task in machine translation (MT) and cross-language information retrieval (CLIR) – translation of toponyms. Due to their linguistic and extra-linguistic nature, toponyms deserve a special treatment.

In general toponyms are studied by toponymy and represent names of places comprising the following types:

- *hydronyms* (names of bodies of water: bays, streams, lakes, lagoons, oceans, ponds, seas, etc., e.g. *Thames* as a river);
- *oronyms* (names of mountains, cliffs, craters, rocks, points, etc., e.g. *Bexhill* as a mountain);
- *geonyms* (general names for streets, squares, lines, avenues, paths, alleys, roads, embankments, etc.);
- *oeconyms* (names of populated places: an administrative division, country, city, town, house or other building).

The first part of the paper overviews the concept and nature of toponyms along with existing toponym translation strategies (TS). The second part of it focuses on the developed and implemented English-Latvian toponym MT approach, including a description of TSs and linguistic toponym translation patterns (LTTP).

2 Concept and Nature of Toponyms

After Geoffrey Leech (1981) we can accept a special status of toponyms as proper names without a conceptual meaning as we cannot perform any componential analysis for them. However, we cannot but admit the fact that many toponyms are at least meaningful etymologically, e.g, *Cambridge* – bridge over the river *Cam* (Leidner, 2007), and, as Leidner pointed out, this etymology might or might not be apparent to a speaker. This feature makes toponyms difficult for processing.

Proceedings of the 13th Annual Conference of the EAMT, pages 81–87,
Barcelona, May 2009

Besides, toponyms are not unambiguous. Leidner (2007) describes three types of the toponymical ambiguity:

- *morpho-syntactic* ambiguity: a word itself may be a toponym or may be a common noun in a language, e.g. *Hook* as the populated place in the UK versus *hook* as a common noun;
- *referential ambiguity*: a toponym may refer to more than one place of the same type, e.g. *Riga* as the populated place and the capital of Latvia and *Riga* as the populated place in the USA, state Michigan;
- *feature type ambiguity*: a toponym may refer to more than one place of different type, e.g. *Tanfield* as the populated place and the castle in the UK, *Gauja* as the populated place and the river in Latvia.

Moreover, there is another type of the toponymical ambiguity to be addressed, that is the so-called *eponymical ambiguity* when names of places are named after people or deities, e.g. *Vancouver* after George Vancouver. In addition, the same place is often known by different names – *endonyms* (names of places used by inhabitants, self-assigned names) and *exonyms* (names of places used by other groups, not locals) as in the Leidner's (2007) example with *Praha* for its inhabitants and *Prague* for English.

Furthermore, metonymy also contributes to the issue. This linguistic phenomenon was studied from the toponymical point of view by Markert and Nissim (2002). The authors stated that the metonymic use of toponyms is regular and productive, can reach up to 17% of all of toponyms as it was proved by the example of the English language, and the most frequent and conventional case of the toponymical metonymy is as in the "*government of ...*" pattern, e.g. "*Latvia announced ...*" means "*the government of Latvia announced ...*".

Finally, toponyms are changed frequently since they themselves and the places they refer to are not constant. Therefore, when dealing with toponyms it is also very important to take into consideration historical and cultural facts.

The abovementioned linguistic and extra-linguistic features make toponym processing difficult, e.g. resolution, retrieval, and especially translation.

3 Toponym Translation Strategies and Approaches

Toponyms can be referred to *named entities* (NE) which comprise all types of proper names, including toponyms themselves, anthroponyms, and temporal expressions. To translate an NE one should choose a TS which depends on the type of the NE unit (Babych and Hartley, 2003), i.e. to translate a toponym we should know its type that assigns a TS to be applied to this toponym. Common TSs for toponyms, as a particular type of NEs, are the following (Babych and Hartley, 2004):

- *transference* strategy, i.e. do-not-translate;
- *transliteration* strategy, i.e. phonetic or spelling rendering;
- *translation* strategy *per se*, i.e. do-translate;
- *combined* strategy, i.e. applying more than one from the abovementioned three strategies.

The transference strategy with the do-not-translate list is often used for translation of toponyms which do not need any rendering at all and are often left not translated, e.g. organization names (Babych and Hartley, 2003).

The topic of transliteration has been studied for several languages, mostly for non-Latin spelling, and many techniques have been proposed. The most common transliteration techniques are phoneme-based and grapheme-based (Zhang et al., 2004). The phoneme-based approach (Knight and Graehl, 1998; Meng et al., 2001; Oh and Choi, 2002; Lee and Chang, 2003) implies a conversion of a source language word into a target language word via its phonemic representation, i.e. grapheme-phoneme-grapheme conversion. The grapheme-based technique converts a source language word into a target language word without any phonemic representation (grapheme-grapheme conversion) (Stalls and Knight, 1998; Li et al., 2004).

Most of toponym translation approaches are data-driven (see, e.g. Meng et al., 2001; Al-Onaizan and Knight, 2002; Sproat et al., 2006; Alegria et al., 2006; Wentland et al., 2008) since they deal with widely used languages which have enough linguistic resources for development. Taking into account an under-resourced status of the Latvian language with few available corpus resources, especially parallel bilingual corpora, a

rule-based approach was proposed for English-Latvian toponym translation.

4 Implementation of English-Latvian Toponym Translation

Strategies and techniques for English-Latvian toponym MT have not been studied previously, and the existing literature describes general principles of rendering of the English proper names, mostly anthroponyms, into Latvian.

We studied three main issues of English-Latvian toponym MT:

- orthographic, phonetic and grammatical differences between the two languages;
- possible toponym translation strategies for this translation direction;
- possible linguistic toponym translation patterns for this translation direction.

Although English and Latvian are the Indo-European languages and share some grammatical features, they have a lot of differences since English belongs to the Germanic language group while Latvian belongs to the group of the Baltic languages; English is an analytical language in contrast to the synthetic Latvian language with a rich set of inflections and some specific orthographic features such as diacritics. The lack of the orthographic and phonetic convergence in English (26 letters to 44 phonemes), historical changes and traditions in spelling, origin language of a toponym, and ambiguity, as well as the lack of the Latvian linguistic resources for the study were the main difficulties we faced. We also studied the peculiarities of Latvian toponymic units to ensure they correspond to the Latvian grammar and orthography rules, e.g.:

- Latvian names are inflected;
- Latvian names cannot be spelled with double consonants, except *ll*, *mm* or *nn* under certain conditions;
- Latvian multi-word units can be translated in several ways, however, a compound is preferable if it allows to reconstruct a source toponymic unit (Ahero, 2006).

4.1 Source String Normalisation

Translation of a toponymic unit is divided into three steps: source string normalisation, translation, i.e. application of TS and LTTP, and target string normalisation according to the Latvian grammar and orthography rules.

Source string normalisation includes the following sub-processes:

- all tabs and double space characters, including the beginning of a string, are normalized to single space characters;
- the so-called "zero-fertility words" (Al-Onaizan and Knight, 2002) of English are normalized to zero-translations in Latvian, e.g. definite article *the* is omitted;
- hyphenated words are normalized to non-hyphenated ones;
- some abbreviations are expanded to full words, e.g. *St.* to *Saint*;
- signs, if possible, are changed to words, e.g. *&* to *and*;
- punctuation marks are normalized to zero translations.

4.2 Translation: English-Latvian Toponym Translation Strategies

Transference strategy is applied to unprocessed toponymic units which are not described by any of LTTPs.

Transliteration strategy is language dependent (Karimi et al., 2007) and for the English-Latvian language pair transliteration is a non-trivial task due to differences in grammar, orthography and sound systems of both languages. Moreover, there are a lot of exceptions (see Castañeda-Hernández, 2004 about general toponym translation problem). English-Latvian transliteration strategy is based on the grapheme-to-grapheme approach, which implies direct mapping of the English letter sequences into the Latvian ones, formalized in a set of transliteration rules. All foreign names (those of non-English origin) are rendered according to the English pronunciation standards. The main principle is the possibility to reconstruct a source toponymic unit (Ahero, 2006).

The set of English-Latvian transliteration rules consists of about 110 transliteration patterns describing English-Latvian grapheme-to-grapheme correspondences. The result of transliteration may vary, as there can be several ways of rendering the English letter combinations into the Latvian ones. Several cases of variety are described by transliteration patterns, e.g. *-c-* stands for *-k-* before consonants (except *-h-*), and *-a-*, *-o-*, *-u-*, for *-s-* before *-i-*, *-e-*, *-y-*, and for *-č-* in the combination with *-h-*.

Translation strategy *per se* is also applied to English-Latvian toponym translation. In some cases toponyms are not transferred or translite-

rated, but translated into Latvian, e.g. multi-word units *East Anglian Heights*, *North West Highlands* are translated into Latvian as *Austrumanglijas augstiene, Ziemeļskotijas kalnāji* correspondingly, while one-word units are transliterated, as a rule. Though, transliteration strategy can be also applied to multi-word units in parallel with translation one which is usually infrequent and conventional.

Toponym TSs are closely related with LTTPs and are language dependent. Therefore, *combined* strategy is also used when treating different types of toponyms.

4.3 Translation: Linguistic Toponym Translation Patterns

When translating a toponymic unit, dictionary-based translation is applied first. Most of popular toponyms, such as names of countries and capitals, seas and oceans, are translated using an English-Latvian dictionary, e.g. *Lisbon – Lisabona*, *Brussels – Brisele*, *Cologne – Ķelne*, *Antwerp – Antverpene*, *Great Britain – Lielbritānija*, *Atlantic Ocean – Atlantijas okeāns*. If a toponym is an out-of-vocabulary (OOV) word then one of the LTTPs is applied.

To determine possible LTTP we studied a list of 10,000 toponyms from Geonames (all toponyms were UK-related) and analyzed 59 toponym types.

Generally, LTTPs are the ways source toponymic units are rendered into target toponymic units. LTTPs can be of two types: in-word patterns and multi-word patterns. The in-word LTTP is a word transformation model, based on English-Latvian transliteration rules, including the most frequent prefixes, suffixes, and letter combinations. There are about 300 in-word LTTPs described, for example: *new-* to *ņū-*, *deep-* to *dīp-*, *mc-* to *mak-*, *-worth* to *–vērt, -islet* to *–ailet*, etc.

Multi-word LTTPs involve three TSs. Translation strategy S_1 is based on transliteration rules. Translation strategy S_2 performs the combination of the first TS and the insertion of a nomenclature word, e.g. *Bebington* (as a railroad station) – *Bebingtonas* ***stacija***. If a nomenclature word is included in a source toponymic unit, as it is in the pattern S_3, it is either translated (*Newton* ***Point*** *- Ņūtona* ***zemesrags***, *Gog Magog* ***Hills*** *- Gogmagogu* ***kalni***) or transliterated (*Green* ***Isle*** *– Grīn****ail****a, North East* ***Coast*** *– Nortīst****kost****a*) in a target language. We described 40 nomenclature words that are translated under certain conditions. Auxiliary words, such as prepositions, are also either translated or transliterated, e.g. *Horse* ***of*** *Copinsay – Hors****of****kopinsejs* (transliteration), *Milford* ***upon*** *Sea - Milforda* ***pie*** *jūras* (translation).

Examples of LTTP are presented in Table 1. X_n is a toponymic unit in a source language, S_n is a translation strategy applied, Y_n is a toponymic unit in a target language, and $P_n\{X_n, S_n, Y_n\}$ is a corresponding LTTP.

4.4 Target String Normalisation

Target string normalisation modifies a toponymic unit according to the rules of the Latvian grammar and orthography, e.g. all populated places are feminine gender (see P1): *Newcastle* → *Ņūkāsl****a*** which is indicated by the ending *–a* (feminine, singular nominative).

English Toponym X_n	**Translation Pattern P_n**	**Translation Strategy S_n**	**Latvian Toponym Y_n**
$P_1\{X_1, S_1, Y_1\}$			
X1: N *Knocklayd*	P1: N → N	S1: transliteration	Y1: N masculine singular *Nokleids*
$P_2=\{X_1, S_1, Y_2\}$			
X1: N *Newcastle*	P2: N → N	S1: transliteration	Y2: N feminine singular *Ņūkāsla*
$P_3=\{X_1, S_2, Y_3\}$			
X1: N *Bebington*	P3: N → N + N	S2: transliteration + nomenclature word	Y3: N feminine singular genitive + N *Bebingtonas stacija*
$P_4=\{X_2, S_1, Y_2\}$			
X2: N's + N *Bishop's Stortford*	P4: N's + N → N	S1: transliteration	Y2: N feminine singular *Bišopsstortforda*

$P_5=\{X_3, S_1, Y_2\}$			
X3: N + N's + N *St. Bishop's Town*	P5: N + N's + N → N	S1: transliteration	Y2: N feminine singular *Sentbišopsatauna*
$P_6=\{X_4, S_1, Y_2\}$			
X4: N + N *Bishop Auckland* *North Ronaldsay*	P6: N + N → N	S1: transliteration	Y2: N feminine singular *Bošopoklenda* *Nortronaldseja*
$P_7=\{X_5, S_1, Y_2\}$			
X5: A + N *South Ribble, Green Isle*	P7: A + N → N	S1: transliteration	Y2: N feminine singular *Sautribla* *Grīnaila*
$P_8=\{X_6, S_3, Y_4\}$			
X6: N + P + N *Milford upon Sea* *Stratford upon Avon*	P8: N + P + N → N + P + N	S3: transliteration + translation	Y4: N feminine singular genitive + P + N *Milforda pie jūras,* *Stradforda pie Avona*
$P_9=\{X_6, S_1, Y_5\}$			
X6: N + P + *Longville in the Dale*	P9: N + P + N → N + N	S1: transliteration	Y5: N feminine singular genitive + N feminine singular locative *Longvila Deilā*
$P_{10}=\{X_7, S_1, Y_2\}$			
X7: A + A + N *North East Coast*	P10: A + A + N → N	S1: transliteration	Y2: N feminine singular *Nortīstkosta*
$P_{11}=\{X_8, S_2, Y_3\}$			
X8: N + C + N *Sandal & Agbrigg*	P11: N + C + N → N + N	S2: transliteration + nomenclature word	Y3: N feminine singular genitive + N *Sendalendagbrigas stacija*
$P_{12}=\{X_4, S_3, Y_6\}$			
X4: N + N *Newton Point*	P12: N + N → N + N	S3: transliteration + translation	Y6: N masculine singular genitive + N *Ņūtona zemesrags*
$P_{13}=\{X_6, S_1, Y_1\}$			
X6: N + P + N *Horse of Copinsay*	P:13 N + P + N → N	S1: transliteration	Y1: N masculine singular *Horsofkopinsejs*
$P_{14}=\{X_7, S_3, Y_7\}$			
X7: N + N + N *Gog Magog Hills*	P14: N + N + N → N + N	S3: transliteration + translation	Y7: N masculine plural genitive +N *Gogmagogu kalni*

"Table 1. Examples of English-Latvian Linguistic Toponym Translation Patterns."

5 Evaluation and Limitations

The current MT evaluation theory and practice lacks in evaluation methods for toponym translation task. One of the reasons could be that it is not clear what the correct toponym translation is, since results may vary and more than one target toponymic unit is acceptable. As a result, scores calculated with a single target variant will underestimate translation accuracy. Moreover, human translations are often inaccurate as well.

Existing English-Latvian MT systems[2] do not implement any OOV algorithms to translate toponymic units. Thus, we had no possibility to

[2] English-Latvian Pragma Expert: www.acl.lv, English-Latvian Google: http://translate.google.com, English-Latvian Tilde http://www.tilde.lv/English/portal/go/tilde/3777/en-US/DesktopDefault.aspx (November, 2008)

compare our algorithm with other MT performance.

For evaluation purposes we compared translation results of our translation module with reference (human) translations from two bilingual dictionaries. 330 English toponymic units of different types with Latvian translation equivalents were manually extracted from dictionaries (180 one-word units and 150 multi-word units) and processed with our OOV toponym translation algorithm. To evaluate translation results we set the following scores:

- if the translation result coincides with the corresponding linguistic toponym translation pattern then the translation is *accurate* and the score is 1;
- if the translation result deviates from the corresponding linguistic toponym translation pattern then the translation is *inaccurate*, and the score is 0,5 for one error and 0 for more errors.

We accept variants as they were also described by LTTPs (in transliteration rules). As a result, the accuracy of translation is 67% on the whole test set, 58% on the set containing one-word toponymic units, and 81% on the multi-word test set.

6 Conclusions and Future Work

We have described the pattern-based toponym translation approach developed for the English-Latvian language pair. We studied the concept and nature of toponyms and several linguistic and extra-linguistic issues, such as ambiguity, cultural and historical changes and others. We also studied different types of toponyms in the context of the overall task of toponym MT.

In the present paper we have overviewed two stages of toponym translation processing: dictionary-based and OOV toponym translation. The latter is divided into three steps: source string normalisation, translation and target string normalisation. The focus of the paper is on detailed description of OOV toponym processing: possible translation strategies and linguistic toponym translation patterns with examples and evaluation results.

We can conclude that for the implemented rule-based approach there is much room for possible improvements, and evaluation results prove this statement. The main reason, why toponym processing is such a challenging task for MT, is the necessity of knowledge of toponym rendering rules, variety of languages as well as a considerable amount of history and culture (Castañeda-Hernández, 2004). It is impossible to formalize this process completely and it is obvious that there can be mistakes in automated translation of toponymic units.

Corpus-based approach has not been applied in this research due to the lack of monolingual and bilingual linguistic resources. However, the issue of compiling a corpus of toponym-referenced texts for the Latvian language is being studied. We also plan to study the issue of multilingual cross-language toponym MT and application MT strategies to other languages (especially Cyrillic or other non-Latin scripts).

Acknoledgement

The research reported was partly funded by the TRIPOD project (TRI-Partite multimedia Object Description)[3] supported by the European Commission under the contract No. 045335.

We would like to thank Lars Ahrenberg, Lars Borin and Raivis Skadiņš for discussions and comments.

References

Antonija Ahero. 2006. *English Proper Name Rendering into the Latvian Language* (Angļu Īpašvārdu Atveide Latviešu Valodā). Zinātne, Rīga.

Iñaki Alegria, Nerea Ezeiza, Izaskun Fernandez. 2006. Named entities translation based on comparable corpora. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Workshop on Multiword expressions in a Multilingual Context*, Italy. Pp.1-8.

Yaser Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. *Proceedings of the 40th Meeting of the Association for Computational Linguistics*, USA. Pp.400-408.

Bogdan Babych and Anthony Hartley. 2003. Improving Machine Translation Quality with Automatic Named Entity Recognition. *Proceedings of the 7th European Association for Machine Translation Workshop Improving machine translation through other language Technology Tools*, Hungary. Pp.1-8.

Bogdan Babych and Anthony Hartley. 2004. Selecting Translation Strategies in MT using Automatic Named Entity Recognition. *Proceedings of the 9th European Association for Machine Translation*

[3] http://tripod.shef.ac.uk/

Workshop Broadening horizons of machine translation and its applications, Malta. Pp.18-25.

Gilberto Castañeda-Hernández. 2004. Navigating through Treacherous Waters: The Translation of Geographical Names. *Translation Journal*, 8(2): [electronic resource]: http://accurapid.com/journal/28names.htm#1

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2007. Collapsed consonant and vowel models: new approaches for English-Persian transliteration and back-transliteration. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Czech Republic. Pp.648-655.

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

Chun-Jen Lee and Jason S. Chang. 2003. Acquisition of English-Chinese Transliteration Word Pairs from Parallel-Aligned Texts using a Statistical Machine Translation Model. *Proceedings of Human Language Technologies – The North American Chapter of the Association for Computational Linguistics Workshop: Building and Using parallel Texts Data Driven Machine Translation and Beyond*, Canada. Pp.96-103.

Geoffrey Leech. 1981. *Semantics. The Study of Meaning*. 2nd edition. Penguin, London, England, UK.

Jochen L. Leidner. 2007. *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names.* PhD thesis. Institute for Communicating and Collaborative Systems School of Informatics, University of Edinburgh.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. *Proceedings of the 42nd Annual Meeting on association for Computational Linguistics*. Spain. Pp.159–166.

Katja Markert and Malvina Nissim. 2002. Towards a corpus annotated for metonymies: the case of location names. *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, France. Pp.1385-1392.

Helen M. Meng, Wai-Kit Lo, Berlin Chen, and Karen Tang. 2001. Generate Phonetic Cognates to Handle Named Entities in English-Chinese cross-language spoken document retrieval. *Proceedings of Institute of Electrical and Electronics Engineers Automatic Speech Recognition and Understanding Workshop*, Italy.

Jong-Hoon Oh and Key-Sun Choi. 2002. An English-Korean Transliteration Model Using Pronunciation and Contextual Rules. *Proceedings of the 19th International Conference on Computational Linguistics*, Taiwan, 1:1-7.

Richard Sproat, Tao Tao, and Cheng-Xiang Zhai. 2006. Named entity transliteration with comparable corpora. *Proceedings of the 44th Annual meeting of the Association for Computational Linguistics*, Australia. Pp.73-80.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating Names and Technical Terms in Arabic Text. *Proceedings of the Coling / Association for Computational Linguistics Workshop on Computational Approaches to Semitic Languages*, Canada. Pp.365-266.

Wolodja Wentland, Johannes Knopp, Carina Silberer, and Matthias Hartung. 2008. Building a Multilingual Lexical Resource for Named Entity Disambiguation, Translation and Transliteration. *Proceedings of the 6th Language Resources and Evaluation Conference*, Morocco.

Min Zhang, Haizhou Li, and Jian Su. 2004. Direct Orthographical Mapping for Machine Transliteration. *Proceedings of the 20th International Conference on Computational Linguistics*, Switzerland.

An Environment for Named Entity Recognition and Translation

Filip Graliński
Adam Mickiewicz University
filipg@amu.edu.pl

Krzysztof Jassem
Adam Mickiewicz University
jassem@amu.edu.pl

Michał Marcińczuk
Wrocław University of Technology
marcinczuk@gmail.com

Abstract

We present an environment for the recognition and translation of Named Entities (NEs). The environment consists of a new formalism for the Named Entity Recognition and Translation (NERT), a parsing mechanism that reads the rules, recognizes Named Entities in given texts and suggests their translation, as well as a set of tools for the evaluation. We suggest a method for the evaluation of (sets of) NERT rules that uses raw (not annotated) bilingual corpora.

1 Introduction

The practical goal of our studies has been to develop a mechanism for correct processing of Named Entities in Machine Translation (MT) systems. Vilar et al (2006) claim that incorrect recognition of NE is responsible for approximately 10% of errors made by MT programs. The authors' experience in the field of MT (e.g. Jassem, 2004; Junczys-Dowmut and Graliński, 2007) tells that incorrect treatment of Named Entities is responsible for most serious errors made by MT programs (i.e. errors that make output incomprehensible to human readers). The research aims at improving the quality of Machine Translation by finding a robust solution for processing NEs in MT systems.

The paper is organized as follows:

In Section 2 we describe the issue of Named Entity Recognition (NER). In Section 3 we show the importance of robust NER solutions for Machine Translation. In Section 4 we present a formalism for the description of NERT rules. In Section 5 we show some examples of rules compatible with the grammar. In Section 6 we discuss the problem of NERT evaluation and suggest an evaluation method that does not require a bilingual corpus to be annotated. We end with the reference to future work in Section 7.

2 Named Entity Recognition

Named Entity Recognition consists in automatic determination of continuous fragments of texts (called Named Entities) which refer to information units such as persons, geographical locations, names of organizations, dates, percentages, amounts of money, locations in texts. A NER module is usually expected to provide a markup on boundaries and types of included NEs.

Here is an example of such a markup from Mikheev (1999), cited also in Nadeau (2007):

```
On <Date>Jan 13th</Date>,
<Person>John Briggs Jr</Person>
contacted <Organization>Wonderful
Stockbrockers Inc</Organization>
in <Location>New York</Location>
and instructed them to sell all
his shares in
<Organization>Acme</Organization>.
```

NER is recognized as a field of Natural Language Processing since 1995. The sixth Message Understanding Conference (Grishman and Sundheim, 1996) is usually considered a starting point of the NER history.

First attempts in the field consisted in creation of handcrafted rules (Rau, 1991; Ravin and Wacholder, 1996). In recent years, this idea has been driven out by machine learning techniques. They include:

supervised learning – NER process is learned automatically on large text corpora and then supervised by a human (Asahara and Matsumoto, 2003; McCallum and Li, 2003)

unsupervised learning – NER process is not supervised; instead, existing semantic lexical databases such as WordNet are consulted automatically (Alfonseca and Manandhar, 2002).

semisupervised learning – this "involves a small degree of supervision, such as a set of seeds, for starting the learning process" (Nadeau, 2007).

Proceedings of the 13th Annual Conference of the EAMT, pages 88–95,
Barcelona, May 2009

The survey of NER solutions is well presented by Nadeau and Sekine (2007).

Some research in the field of NER has been done for the Polish language by Piskorski (2005). The author developed a rule-based formalism for the recognition of named entities in Polish texts and handcrafted a set of NER rules for Polish.

3 Named Entity Recognition in MT

Vilar et al (2006) classify errors made by MT systems. The most general classes of errors are: Missing Words, Word Order, Incorrect Words, Unknown Words, Punctuation. The classification does not include the class (or subclass) named Wrong NE Translation. This is probably due to the fact that errors of this type are hard to classify, which in turn is a result of the fact that incorrect NE translation may cause any of the following errors: Word Order, Incorrect Words, Unknown Words. On the other hand, while examining the percentage of error types in various documents, later in the same paper, the authors introduce the error class "Named Entity" and claim that approximately 10% of MT errors may be classified as belonging to the class.

Our experience with the MT system Translatica (www.poleng.pl, www.translatica.pl) shows an even stronger need for correct recognition and translation of NEs. This is particularly important for free-order languages (like those of the Slavonic origin). Incorrect recognition of NE boundaries results in incorrect syntactic analysis of the sentence, which may be shown by the following example:

```
Podała rękę <Person-dative> Pani
Prezes Justynie Kowalskiej</Person-
dative>.
```

The above correct NE recognition leads to the correct translation:

```
She gave a hand to Mrs. Justyna
Kowalska, Chairperson.
```

Suppose that the same source sentence is erroneously processed by an imperfect NER module as:

```
Podała rękę <Person-nominative>Pani
Prezes</Person-nominative> <Person-
dative>Justynie Kowalskiej</Person-
dative>.
```

The above incorrect recognition would lead to the incorrect translation:

```
A chairperson gave a hand to Justy-
na Kowalska.
```

(It is worth noting that NER results for synthetic languages should contain linguistic information, such as case (e.g. Person-dative) to allow correct syntactical parsing.)

Basic research on Named Entity Recognition and Translation has focused on the paradigm of Statistical MT. The ideas presented by Huang (2005), Huang et al (2005) and Al-Onaizan & Knight (2002) aim at statistical methods to collect bilingual lexicons of Named Entities.

We are of the opinion that the purely statistical approach does not solve the NERT robustly because, unlike ordinary words, Named Entities are characterized by fewer repetitions. Instead, we propose the following approach:

1) NERT rules are first handcrafted according to a given formalism;
2) A testing environment allows automatic evaluation of the impact of the rules on translation quality;
3) The rules are enhanced by means of semi-supervised learning.

Babych and Hartley (2003) put forward a hypothesis that MT quality could be significantly improved if NER results were incorporated into MT systems. They carried out an experiment that consisted in incorporating the results of the GALE project into existing commercial MT systems (Systran, Reverso, ProMT). The results of NER tools were manually included into the MT system as Do-Not-Translate lists. The authors reported improvement in the quality of translation.

In 2004, we tried to follow this idea for our MT system, Translatica. Soon, we discovered that Do Not Translate idea is not sufficient for our needs. Thus, we extended the formalism for the rules so that it would allow for translation of (parts of) Named Entities. We also added types of recognized NEs, as they were needed for semantic analysis. Equipped with that, we tried to incorporate the NER rules into our translation system. The results were disappointing (improvement of translation quality in some areas was offset by deterioration in others) and we decided to give up the idea and wait for further development in the area of NER.

A paper by Piskorski (2005) gave some hope of attacking the problem again. However, a complex formalism suggested there in our opinion makes it difficult for linguists or machine learning algorithms to create the rules.

The Spejd formalism invented by Przepiórkowski (2008), intended basically for shallow parsing of a text (not necessarily for the needs of MT), gave us new hopes for handling the problem. Our formalism, presented in the Appendix and described in Section 4, is the extension of the Spejd notation. Our engine, intended for named entity

recognition and translation (NERT), based on the formalism, was written from scratch.

4 NERT grammar

In this section, we discuss the components of the NERT grammar. Its detailed description is given in Appendix.

4.1 NERT definitions

NERT definitions aim at simplifying rules by using labels (in curly brackets) instead of longish expressions, e.g.:

```
UpperPL=[A-ZĄĆĘŁŃÓŚŹŻ]
LowerPL=[a-ząćęłńóśźż]

# Polish word starting with
# a upper-case letter:
ProperPL={UpperPL}{LowerPL}*

# Polish first name:
FirstNamePL
=<{ProperPL};sem=first_name>

# Sequence of any number of first
# names and a ProperPL
PersonPL={FirstNamePL}+ <{ProperPL}>
```

4.2 Match part of the rule

Any NERT rule consists of the match part and the action part. The match part consists of the main matching pattern and some optional context patterns:

Before: pattern

Imposes the conditions on the context preceding the match in the same sentence – directly or indirectly.

Left: pattern

Imposes the conditions on the context preceding the match directly, in the same sentence.

Match: pattern

Imposes the conditions on the matching pattern.

Right: pattern

Imposes the conditions on the context following the match directly, in the same sentence.

After: pattern

Imposes the conditions on the context following the match in the same sentence – directly or indirectly.

Exists: pattern

Imposes the conditions on the context occurring anywhere in the same sentence.

4.3 Action part of the rule

The action part of the rule creates the translation for the recognized NE. The translation is executed by copying or modifying groups of the NE, or adding new texts to the equivalents. There are two types of actions in the NERT formalism:

prepend adds "sure" translation of the recognized entity;

append adds "unsure" translation of the recognized entity.

The need for distinguishing between ***append*** and ***prepend*** is that some NEs might be alternatively processed by other translation modules. In such a case ***prepend*** gives priority to the NER module, whereas ***append*** leaves priority to other modules.

4.4 Group Ordering

Each group that occurs in the match part of the rule is assigned an ordering consecutive integer.

Suppose that the analyzed text contains a string *pani Prezes Justynie Marii Kowalskiej* (dat. *Mrs. Justyna Maria Kowalska, Chairperson*). The match part of the rule for such NEs may have the following form:

```
Match: <base~pani> <{ProperPL}>
{FirstNamePL}+ <{ProperPL}>
```

The recognized groups are then ordered as follows: *pani* – 1, *Prezes* – 2, *Justynie Marii* – 3, *Kowalskiej* – 4.

Group ordering integers are referred to in the action part of rules.

4.5 Modifiers

Modifiers operate on the recognized groups:

t – translate the group (use the lexicon)

nom, gen, dat, acc, instr, loc – replace the group with its appropriate inflected case

s[[(**+**|**-**)*Num*][**,**][(**+**|**-**)*Num*]**]** – cut characters from the given range of the group, e.g. **s[-1**] cuts the last character

u – uppercase the first letter of every token in the group

The ordering integers for recognized groups are preceded by '\', e.g.:

\1:t – translate the first group of the recognized entity (use lexicon)

\3:nom – replace the third group of the recognized entity with its nominative case.

For instance, the following action translates the entity *pani Prezes Justynie Kowalskiej* (assuming that each word matches one group) into *Mrs. Justyna Kowalska, Chairperson*:

```
prepend(Mrs. \3:nom \4:nom, \2:t)
```

The following action translates *2008r* (*r* stands for *rok* = *year*) into *2008*:

```
prepend(\1:s[-1])
```

4.6 Commands

Commands set the values of attributes of the translated NE. For example, for setting the semantic class of a recognized NE `sem=` command should be used:

```
prepend(Mrs. \3:nom \4:nom, \2:t;
     sem=person)
```

5 Examples of NERT rules

5.1 Corporation recognition rules

Some named entities denoting corporations may be recognized by their specific endings, such as "S.A" (English: "jsc"). A simple rule may look like this:

```
Match: <{ProperPL}>+ <S.A.>
Action: prepend(\1 \2)
```

This would suffice for correct recognition and translation of the following texts:

```
Indykpol S.A.
Bank Handlowy S.A.
```

However, the above rule would not translate correctly the following text:

akcje **Banku Handlowego S.A.**
(= shares of Bank Handlowy S.A.)

Here, the named entity (in bold) should not be just copied. Instead, it should be transformed into the nominative case (*Bank Handlowy S.A.*).
The rule needs adjustment:

```
Match: <{ProperPL}>+ <S.A.>
Action: prepend(\1:nom \2)
```

This solution will still leave open the problem of words starting with an upper-case letter that precede named entities, as in the following two texts:

Wiceprezes Zarządu Banku ***PKO SA;***
Zwyczajnego Walnego Zgromadzenia Akcjonariuszy **INDYKPOL S.A.,**

The underlined fragments lie beyond the scope of the named entities.
An exemplary rule may look like this:

```
CORP_AFFIX=wiceprezes|akcjonariusz|
zarząd|other words used in terms
denoting (members of) company bod-
ies
CORP_NAME=<{ProperPL};base!~{CORP_A
FFIX}>+
CORP_SUFFIX=S.A.
Match:{CORP_NAME} <{CORP_SUFFIX}>
Action: prepend(\1:nom \2; sem=or-
ganization)
```

The name of the organization may include a name of a city:

```
Bank Przemysłowo-Handlowy w
Krakowie SA.
```

An appropriate NERT rule looks like this:

```
Match: {CORP_NAME} <w> <sem=city>
<{CORP_SUFFIX}>
Action: prepend(\1:nom w \3 \4;
sem=organization)
```

5.2 Temporal expressions

The presented NERT mechanism allows for recognition and translation of temporal expressions. Here are some examples:

```
Match: <1> <base~kwartał> <[0-9]{4}r\.>
Action: prepend(1st quarter of \3:s[-2];
sem=time_period)
```

Example: *1 kwartale 2008r. = 1st quarter of 2008*

```
Match: <4> <kw> <[0-9]{4}>
Action: prepend(4th quarter of \3;
sem=time_period)
```

Example: *4 kw 2010 = 4th quarter of 2010*

```
Match: <base~{MonthPL}> <[0-9]{4}r\.>
Action: prepend(\1:t \2:s[-2];
sem=month)
```

Example: *lutego 1986r.* (gen.) = *February 1986*

```
Match: <[0-9]{1,2}> <base~{MonthPL}>
<[0-9]{4}> <r\.>
Action: prepend(\2:t \1, \3; sem=date)
```

Example: *1 czerwca 2007 r. = June 1, 2007*

5.3 Legal terms

In the machine translation of legal texts, one of the particular problems is the processing of references to act articles, e.g.

```
Original text: Podstawa prawna:
Art. 56 ust. 1 pkt 1 Ustawy z dnia
29 lipca 2005
Expected translation: Legal
grounds: Art. 56.1.1 of the Act of
29 July 2005
```

A NERT rule that processes the above Named Entity (reference to a location in a document) looks like this:

```
Match: <Art\.> <{NUM}> <ust\.>
{NUM} <pkt> <{NUM}> <[Uu]stawy> <z>
<dnia> <[0-9]{1,2}>
<base~{MonthPL}> <[0-9]{4}>
Action: prepend(Art. \2.\4.\6 of
the Act of \10 \11:t \12; sem=docu-
ment)
```

6 Evaluation

In the evaluation of NER systems two measures are referred to most often: precision and recall (sometimes they are merged in one measure, e.g. F-score). Precision is the ratio of the correct guesses to the number of all guesses, recall is the ratio of the correct guesses to the actual number of NEs in the text.

The question is how to treat the partial guesses, for instance the correct recognition of the NE type and the incorrect recognition of the NE boundaries.

There exist two approaches: one approach assigns a point for each correct type recognition

(TYPE) and each correct boundaries recognition (TEXT):

correct TYPE incorrect TEXT – 1 point
incorrect TYPE correct TEXT – 1 point
correct TYPE and correct TEXT – 2 points

(To calculate the recall, the actual number of NEs is multiplied by two).

In the other approach only guesses that are correct both in TYPE and TEXT are assigned a point:

TEXT and TYPE – 1 point
Otherwise – 0 points

See Nadeau (2007) for a more detailed discussion on NER evaluation.

In our opinion, it is crucial for MT goals that the boundaries are recognized correctly. Moreover, we need an additional parameter for the correct translation of NE. Therefore we suggest the following method of scoring for the evaluation of NERT:

incorrect TEXT – 0
correct TEXT correct TYPE incorrect TRANSLATION – 1 point
correct TEXT incorrect TYPE correct TRANSLATION – 1 point
correct TEXT correct TYPE correct TRANSLATION – 2 points

Here is an example of how the suggested NERT evaluation may work:

Podała rękę Pani Prezes Justynie Kowalskiej.

Possible NERT recognitions:

1) Podała rękę Pani Prezes <TYPE: PERSON; TRANSLATION: Justynie Kowalskiej>Justynie Kowalskiej</PERSON>
Score – 0 (correct TYPE, incorrect TEXT, incorrect TRANSLATION)
Recall – 0
Precision – 0
2) Podała rękę Pani Prezes <TYPE: PERSON; TRANSLATION: Justyna Kowalska>Justynie Kowalskiej</PERSON>
Score – 0 (correct TYPE, incorrect TEXT, correct TRANSLATION)
Recall – 0
Precision – 0
3) Podała rękę <TYPE: PERSON; TRANSLATION: Mrs. Chairperson Justyna Kowalska> Pani Prezes Justynie Kowalskiej</PERSON>
Score – 1 (correct TYPE, correct TEXT, incorrect TRANSLATION)
Recall – 0,5
Precision – 0,5
4) Podała rękę <TYPE: PERSON; TRANSLATION: Mrs. Justyna Kowalska, Chairperson>Pani Prezes Justynie Kowalskiej</PERSON>
Score – 2
Recall – 1
Precision – 1

In order to provide such an evaluation of a NERT module one needs to have access to an appropriately annotated corpus.

We have calculated the Precision of our methods in the following way:

1) Handcraft an initial set of NERT rules;
2) Run the NERT mechanism consistent with the rules against a set of approximately 10 000 Polish sentences from legal documents;
3) Select sentences which contain recognized NEs;
4) Divide the resulting set into two equal parts;
5) Evaluate the set of rules against the first half;
6) Adjust the rules;
7) Evaluate the set of rules against the remaining half.

To evaluate the results, three translators have been requested to verify the translation of all entities recognized by the modules. For each of 3160 entities the translators scored their TEXT, TYPE or TRANSLATION by either 1 point (correct) or 0 points (incorrect).

Table 1. shows the Precision calculated in the strict approach: set 1 point for the named entity with all of TYPE, TEXT and TRANSLATION values equal to 1, set 0 otherwise:

#NE	Max score	Actual score	Precision
3160	3160	2413	76,36%

Table 1.

Table 2 shows Precision, which allows for partial scores.

#NE	Text	Type	Translation	Max score	Actual score	Prec.
3160	2853	3002	2515	9480	8370	88,29

Table 2.

Table 3 shows Precision calculated in the method suggested in the paper:

#NE	Max score	Actual score	Precision
3160	6320	5132	81,20%

Table 3.

The above-mentioned method of NERT evaluation has required plenty of human work (it took 6 translators' workdays to estimate our results). We therefore suggest another method for the NERT evaluation – using the METEOR metrics. METEOR (Banerjee, 2005) is the metrics intended for the evaluation of MT algorithms – by comparing their output to reference texts, translated by humans. METEOR is based on BLEU

(Papineni, 2002) but it emphasizes more recall than precision.

The idea has the following merits:

(1) No annotated corpora are needed;

(2) The evaluation may be executed automatically for any selected subset of the NERT rules (including a single rule):

```
1) Take a bilingual "golden standard" corpus of manually translated texts (S | T), the set of all rules ALL, and the set of selected rules SELECTED
2) Translate all sentences from the corpus S:
      2.1 using rules from ALL, obtaining translation T1
      2.2 using rules from the difference: ALL – SELECTED, obtaining translation T2
3. Using METEOR metrics:
    3.1. Compare T1 to T, obtaining METEOR(T1)
    3.2. Compare T2 to T, obtaining METEOR(T2)

4. If METEOR(T1) – METEOR(T2) > F1 (positive threshold)
      then assume SELECTED as useful

   If METEOR(T2) – METEOR(T1) > F2 (negative threshold)
      then assume SELECTED as undesirable
    Otherwise assume SELECTED as unreliable
```

The METEOR evaluation of our preliminary efforts for the whole set of handcrafted Polish-to-English rules are shown in Table 4.

#sentences	avg. score without NERT	#sentences changed with NERT	avg. score with NERT
9794	0.577	1461	0.581

Table 4.

7 Future work

As reported in this paper, the first step of the research has been to create the NERT mechanism and incorporate it into an existing MT system.

The next step would be to create a testing environment, which will allow for the following supervision functionalities:

Rule edition: Edit a rule; Erase a rule; Create an inverted language direction rule.

Rule evaluation: Select a testing text corpus; Use the complete set of rules to test against the golden standard; Use an incomplete set of rules to test against the golden standard; Compare the tests; Use regressive tests.

We will develop the rules for 5 language pairs: Polish-English/French/German/Russian/Spanish.

The seed sets of rules will be hand-crafted. Then the rules will be refined statistically. The testing environment will allow for supervision.

7.1 Statistical rule acquisition

We claim that human translation of NE between languages that use the same alphabet is reliable and therefore we want to use human expertise while creating NERT rules. On the other hand, we would like to benefit from existing bilingual corpora. Therefore we intend to develop statistical methods for the acquisition of NERT rules. These rules will be automatically evaluated against a bilingual corpus (see Section 6) and finally verified by humans.

Our method is similar to the semi-supervised learning used by Nadeau (2007). There, the author manually creates seeds of NE, on which the system learns new Named Entities. We shall create the seed rules. The system will learn new rules statistically.

To clarify the intended algorithm we show how it should work on exemplary definitions and a rule R.

```
CORP_AFFIX=<base~(prezes|akcjonariusz|zarząd)>
CORP_SUFFIX=S.A.
R:
Left: <{CORP_AFFIX}>
Match: {CORP_NAME} <{CORP_SUFFIX}>
```

The algorithm is to identify other, so far unknown, affixes that can occur directly before a company name. A new NERT rule (meta-rule) M is designed, where the affix is replaced by a wild character:

```
M:
 Left: <base~.*>
 Match: {CORP_NAME} <{CORP_SUFFIX}>
```

Rule R is run against a corpus. Suppose R finds the following matches:

```
Prezes Polmos S.A.
Zarząd Citronex S.A
```

The following rules are derived from the meta-rule M and the set of found matches:

```
M1:   Left: <base~.*>
      Match: <Polmos> <S.A.>
M2:   Left: <base~.*>
      Match: <Citronex> <S.A.>
```

Now, M1, M2 are run against the corpus, resulting in new matches, e.g.:

```
Wiceprezes Polmos S.A.
Sekretariat Citronex S.A.
```

This, in turn, results in new NER rules:

```
R1: CORP_AFFIX=<base~(wiceprezes)>
      CORP_SUFFIX=S.A.
R2: CORP_AFFIX=<base~(sekretariat)>
      CORP_SUFFIX=S.A.
```

The ACTION part of the rules is copied from rules prepared by humans.

Acknowledgment

The paper is based on research funded by the Polish Ministry of Science and Higher Education (Grant No 003/R/T00/2008/05).

References

Alfonseca, Enrique and S. Manandhar (2002), Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery. *Proc. Intl. Conference on General WordNet*.

Asahara, Masayuki and Y. Matsumoto (2003), Japanese Named Entity Extraction with Redundant Morphological Analysis. *Proc. Human Language Technology conference - North American chapter of the Association for Computational Linguistics.*

Babych, B., and A. Hartley (2003), Improving Machine Translation quality with automatic Named Entity recognition. *Paper presented at the 7th International EAMT workshop on MT and other language technology tools at the 10th Conference of the European Chapter of the Association for Computational Linguistics EACL 2003, Budapest.*

Banerjee, Satanjeev and Alon Lavie (2005), METEOR: An Automatic Metric For MT Evaluation With Improved Correlation With Human Judgments. Workshop: On Intrinsic And Extrinsic Evaluation Measures For Machine Translation And/or Summarization

Grishman, Ralph and Beth Sundheim (1996), Message Understanding Conference - 6: A Brief History. In: Proceedings of the 16th International Conference on Computational Linguistics, I, 466–471.

Huang F. (2005), Multilingual Named Entity Extraction and Translation from Text and Speech, Ph.D. Thesis. Carnegie Mellon University.

Huang F., Y. Zhang, and S. Vogel (2005), Mining Key Phrase Translations from Web Corpora. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, 483–490.

Jassem K., (2004), Applying Oxford-PWN English-Polish dictionary to Machine Translation, Proceedings of 9th EAMT Workshop, "Broadening horizons of machine translation and its applications", Malta, 26-27 April 2004

Junczys-Dowmut M. and F. Graliński (2007), Using a Treebank Grammar for the Syntactical Annotation of German Lexical Phrases, *Proceedings of 3rd L&T Conference*, Poznań 2007

McCallum, Andrew; Li, W. (2003), Early Results for Named Entity Recognition with Conditional Random Fields, Features Induction and Web-Enhanced Lexicons. In *Proc. Conference on Computational Natural Language Learning*.

Mikheev, Andrei, (1999), A Knowledge-free Method for Capitalized Word Disambiguation. In: *Proc. Conference of Association for Computational Linguistics*.

Nadeau, D. and S. Sekine (2007), A Survey of Named Entity Recognition and Classification. In: Sekine, S. and Ranchhod, E. *Named Entities: Recognition, classification and use*. Special issue of Lingvisticæ Investigationes.

Nadeau, D. (2007), Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision, PhD thesis, University of Ottawa

Al-Onaizan Y. and Knight K. (2002), Machine transliteration of names in Arabic text Full text , *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, Philadelphia, Pennsylvania

Papineni, K., Roukos S., Ward T. and Zhu W.-J. (2002), BLEU: a method for automatic Evaluation of Machine Translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02), pp. 311-318*

Piskorski, (2005), Named-Entity Recognition for Polish with SProUT. in: Leonard Bolc, Zbigniew Michalewicz, Toyoaki Nishida (eds.): *Lecture Notes in Computer Science Vol 3490 / 2005: Intelligent Media Technology for Communicative Intelligence: Second International Workshop, IMTCI 2004, Warsaw, September 13-14. Revised Selected Papers, Pages 122-, Springer-Verlag 10/2005,*

Przepiórkowski A. (2008). *Powierzchniowe przetwarzanie języka polskiego*. Warszawa: Akademicka Oficyna Wydawnicza EXIT.

Rau, Lisa F. (1991), Extracting Company Names from Text. In *Proc. Conference on Artificial Intelligence Applications of IEEE*.

Ravin, Yael and N. Wacholder (1996), *Extracting Names from Natural-Language Text*. IBM Research Report RC 2033

Vilar, D., J. Xu, L.F. D'Haro and H. Ney (2006), Error Analysis of Statistical Machine Translation Output. *Proc. Language Resources and Evaluation conference*

Appendix A. NERT Grammar

```
::nert_file::       (definition)*       # set of definitions
                    (rule)+             # non-empty set of rules

::definition::      Name =pattern       # definition of a pattern

::rule::    Rule(Name)                  # descriptive name of the rule
            [Before: pattern]           # pattern preceding the match in the same sentence (optional)
            [Left: pattern]             # left context of the match (optional)
            Match: pattern              # matching text
            [Right: pattern]            # right context of the match (optional)
            [After: pattern]            # pattern following the match in the same sentence (optional)
            [Exists: pattern]           # pattern in the same sentence as the match (optional)
            Action: action_list         # action evoked if the match is found in the specified context

::pattern::         group( group)*      # group is a sequence of tokens that meet the same conditions

::group::           NertRegExp          # a regular expression that may use a NERT definition in brackets

::group:: <condition(;condition)*>       # set of conditions for the pattern to satisfy
          ( *|+|?|{Num(,Num)})?         # number of consecutive tokens that should satisfy the conditions

::condition::       orth|               # orthographical form of the pattern or
                    base                # canonical form of the pattern being a word or a word phrase
                    (~|!~)NertRegExp    # matches (or not) a NERT regular expression

::condition::       (pos|               # part of speech of the pattern being a word or a word phrase
                    case|               # case
                    num|                # number
                    gen|                # gender
                    deg|                # degree
                    per|                # person
                    sem                 # semantic class
                    )= Value

::action_list::     do(, do)*           # list of actions which transform source text into target text

::do::      prepend(newText [;Num [; Num ]][; command_list]) |  # add "sure" translation
            append(newText [;Num [; Num ]][; command_list])     # add "unsure" translation

::newText::         expression( expression)*

::expression::      Text |              # new text in the translation output
                    derived             # text derived from source match

::derived::         \Num(:modifier+)?           # copy or modify Numth element of the match

::modifier::| nom | gen | dat | acc | instr | loc |      # set the appropriate case
            t |                                          # translate (use lexicon)
            s[[(+|-)Num][,][(+|-)Num]]                   # cut characters from the text
            u                                            # uppercase the first letter

::command_list::  command (, command)*

::command:: (pos | case | num | gen | deg | per | sem) = (@Num | Value)
                        # copy the attribute value from Numth element of the match or set given value

::command:: all = @Num                                   # copy all attributes values from Numth element
```

Name, Text, Value – any text strings
Num – any number
NertRegExp – a regular expression that may use NERT definitions in brackets.

Optimal Bilingual Data for French–English PB-SMT

Sylwia Ozdowska and **Andy Way**
National Centre for Language Technology
Dublin City University
Glasnevin, Dublin 9, Ireland
{sozdowska,away}@computing.dcu.ie

Abstract

We investigate the impact of the original source language (SL) on French–English PB-SMT. We train four configurations of a state-of-the-art PB-SMT system based on French–English parallel corpora which differ in terms of the original SL, and conduct experiments in both translation directions. We see that data containing original French and English translated from French is optimal when building a system translating from French into English. Conversely, using data comprising exclusively French and English translated from several other languages is suboptimal regardless of the translation direction. Accordingly, the clamour for more data needs to be tempered somewhat; unless the quality of such data is controlled, more training data can cause translation performance to decrease drastically, by up to 38% relative BLEU in our experiments.

1 Introduction

Statistical machine translation (SMT) systems are trained on sentence-aligned parallel corpora consisting of translated texts. In the simplest case the translation direction is constant so that one part of the parallel corpus is the translation of the other. In more complex cases, either some texts may have been translated from language A to language B and others the other way round, or more than two languages are involved and both parts were translated from one another or several other languages. This is the case of corpora involving European languages, such as the Europarl corpus (Koehn, 2005)[1] or the Acquis Communautaire corpus (Steinberger et al., 2006)[2], which comprise texts coming from institutions of the European Union. They are amongst the largest and most widely used corpora in SMT.

Typically, given a corpus in language A, its version in language B and an SMT system translating from A to B, SMT training assumes A to be the source language (SL) and B to be the target language (TL) irrespective of the original translation direction or languages involved. In other words, it is assumed that the original SL does not matter when training an SMT system which aims to translate from language A to language B.

Following a brief overview of related work (section 2), we investigate the impact of the original SL with regard to French–English translation. Our experimental objective is to compare training configurations which differ in terms of the original SL by measuring French-to-English and English-to-French translation quality of a state-of-the-art phrase-based SMT (PB-SMT) system. We train four different configurations of the same PB-SMT system based on French–English parallel corpora which differ in terms of the original SL (sections 3 and 4) and carry out translation experiments from French into English and from English into French (section 5). We evaluate each output using standard evaluation metrics, compare the results and present our findings (section 6). We then conclude and give some avenues for future work (section 7).

2 Related work

Although it is a big topic of interest in translation studies, directionality seems to have been almost

[1] http://www.iccs.inf.ed.ac.uk/\~pkoehn/publications/europarl/
[2] http://wt.jrc.it/lt/Acquis/

Proceedings of the 13th Annual Conference of the EAMT, pages 96–103,
Barcelona, May 2009

totally neglected in SMT research. In the context of SMT, the question of directionality is not addressed directly. Instead, Wu and Wang (2007) propose a method for PB-SMT based on a pivot language to translate between languages for which there exist only small amounts of or no parallel data. They show for instance that good translation quality can be achieved when using Greek as pivot to translate from French into Spanish. In the context of translation studies, Teubert (1996) claims that if a text is translated from language A into languages B and C, then the B and C versions are likely to bear more resemblance to A than to each other. More generally, it seems to be acknowledged that translated texts should not be viewed as bidirectional resources (Bowker, 2003).

Therefore, it seems reasonable to think that there might be a correlation between MT quality from language A to language B and the actual "translational status" of languages A and B in the training corpus and the testset. More precisely, our hypothesis is that using data where A is the original SL and B the TL is likely to be the optimal configuration with regard to MT quality from A to B. Conversely, the case where neither A nor B is the original SL, meaning that both are translated from other languages, is expected to be the suboptimal configuration.

In order to test whether this hypothesis holds true, we perform training on four sub-corpora extracted from the Europarl corpus, namely: a) no criterion is imposed on the original SL, b) the original SL is neither French nor English, c) the original SL is French and d) the original SL is English. We then measure translation accuracy according to a range of automatic MT evaluation metrics.

3 Data

3.1 The Europarl corpus

In the experiments we present here, we used an in-house version of the French–English part of the original Europarl corpus.[3] Some manual changes were made to the original files to correct misalignments (*e.g.* extra, empty speaker turns) prior to sentence alignment performed automatically with a technique based on (Gale and Church, 1993). The alignments at sentence level were tagged with information on the original SL.

[3]Thanks to Mary Hearne for providing us with the modified version of the Europarl corpus.

Table 1 gives the spread in terms of number of sentence pairs according to the original SL. It can be seen that out of 1,391,222 French–English sentence pairs appearing in the corpus, only 164,648 were originally translated from French into English and 235,102 the other way round. For 715,090 sentence pairs, the original SL is neither French nor English, meaning that both the French part and the English part of the corpus contain translations from the other 20 source languages represented. Hence translated French and translated English account for at least 50% of the corpus; the original source language is unknown (NONE and EMPTY) for 276,382 sentence pairs.

original SL	sentence pairs
NONE	259540
Enlish	235102
German	201195
French	164648
Dutch	121045
Spanish	84285
Italian	68259
Swedish	56377
Portugese	49183
Greek	43541
Finnish	31334
Danish	25506
EMPTY	16842
Polish	15714
Czech	4613
Hungarian	4589
Slovak	2702
Lithuanian	2034
Latvian	1388
Slovenian	1380
Maltese	996
Estonian	949

Table 1: Repartition according to the original SL in the French–English Europarl corpus

Therefore, the French–English part of the version of the Europarl corpus our experiments are based on is made up of texts where:

- the original SL is French, and hence the English side contains English translated from French;
- or the original SL is English, and hence the French side contains French translated from English;

- or the original SL is neither French nor English, and hence both the French and the English side contains translated French or English.

3.2 Dataset extraction

In order to investigate the influence of the original SL on French–English state-of-the-art PB-SMT, we built four configurations of the same system for each translation direction based on the information on the original SL. Each configuration was built and tested using a French–English dataset (training data and testsets) extracted according to a different criterion as to the original SL. The original SL selection criteria and the contents of the four datasets extracted are described in the following section. The datasets were tokenised and lowercased for the purpose of the experiments. Moreover, only sentence pairs corresponding to a 1-to-1 alignment with lengths ranging from 5 to 40 tokens on both French and English sides were considered. We used 100,000 sentence pairs for training and 500 sentences to test each configuration and measure translation quality.

3.3 Training and test configurations

config-1 No condition is imposed on the original SL, meaning that the French part of the data and its English counterpart contain respectively:

- French translated from ¬English, French translated from English and original French;
- English translated from ¬French, English translated from French and original English.

Table 2 shows the repartition in terms of number of sentence pairs according to the original SL for the training corpus and the testset associated with config-1. It can be seen that both the training corpus and the testset show a similar spread as to the original SL.

config-2 The original SL is neither French nor English, meaning that the French part of the data and its English counterpart contain respectively:

- French translated from ¬English;
- English translated from ¬French.

Table 3 shows the repartition in terms of number of sentence pairs according to the original SL for the training corpus and the testset associated with config-2. Here again the repartition was kept as consistent as possible across the training data and the testset.

original SL	train sentences	test sentences
German	17551	116
English	16635	58
French	15697	47
NONE	12912	98
Dutch	11691	50
Spanish	6260	50
Swedish	4981	22
Italian	3974	22
Portugese	3155	15
Finnish	2772	15
Greek	2458	0
Danish	1914	7

Table 2: Config-1 – training data and testset in terms of original SL

original SL	train sentences	test sentences
German	30467	232
Dutch	21638	115
Swedish	11556	37
Spanish	11265	43
Italian	7497	14
Portugese	5092	23
Finnish	4737	25
Greek	4252	11
Danish	3496	0

Table 3: Config-2 – trainig data and testset in terms of original SL

config-3 The original SL is English, meaning that the French part of the data and its English counterpart contain respectively:

- French translated from English;
- original English.

To evaluate the performance of config-3 for French-to-English translation, we use a portion of the French part of the data (*i.e.* French translated from English) as test and the English part (*i.e.* original English) as reference. English-to-French translation evaluations are based on the same portion of the data; this time, the English part (*i.e.* original English) is used as test and the French part (*i.e.* French translated from English) as reference.

config-4 The original SL is French, meaning that the French part of the data and its English counterpart contain respectively:

- original French;
- English translated from ¬French.

To evaluate the performance of config-4 for French-to-English translation, we use a portion of the French part of the data (*i.e.* original French) as test and the English part (*i.e.* English translated from French) as reference. English-to-French translation evaluations are based on the same portion of the data; this time, the English part (*i.e.* English translated from French) is used as test and the French part (*i.e.* original French) as reference.

In addition to each individual 500-sentence testset, we also constructed one unique testset of 2000 sentences by merging the individual tests. The composition in terms of original SL of the 2000-sentence testset is given in Table 4. Overall evaluations in both translation directions are carried out based on this testset. For French-to-English, the French part is used as test and the English part as reference. For English-to-French, the latter is used as test and the former as reference.

original SL	test sentences
English	558
French	547
German	348
Dutch	165
NONE	98
Spanish	93
Swedish	59
Finnish	40
Portugese	38
Italian	36
Greek	11
Danish	7

Table 4: Test-2000 – repartition according to the original SL

4 Tools

4.1 Alignment and translation

All translation experiments are carried out using standard state-of-the-art techniques. Sentence pairs are first word-aligned using GIZA++ implementation of IBM model 4 in both source-to-target and target-to-source translation directions (Brown et al., 1993; Och and Ney, 2003) for each training set. After obtaining the intersection of these directional alignments, alignments from the union are also inserted; this insertion process is heuristics-driven (Koehn et al., 2003). Once the word alignments are finalised, all word- and phrase-pairs which are consistent with the word alignment and which comprise at most 7 words are extracted. Phrase-pairs are extracted by standard PB-SMT techniques using the Moses system (Koehn et al., 2007). A 5-gram language model is trained with SRILM (Stolcke, 2002) on the English side of the training data for French-to-English translation experiments and on the French side of the training data for English-to-French translation experiments. Finally decoding is carried out with Moses.

4.2 Minimum error rate training

Due to time constraints, we do not perform minimum error rate training (MERT) although it is now well established as a standard technique in PB-SMT (Och and Ney, 2003). Our experimental objective is to compare the relative performance of four configurations of the same system for each translation direction which differ only according to the conditions imposed on the original SL when selecting the dataset they are trained and tested on. We are not interested in the absolute performance each of these configurations achieves individually as far as the experiments presented here are concerned. Although carrying out MERT would probably have led to an increase in translation quality achieved with the different configurations that are tested, we have no reason to think that it would have resulted in a radical change as to their relative performance. However, this assumption needs to be confirmed by further experiments, which are currently ongoing (cf. footnote 4).

4.3 Evaluation

The results of the translation output are evaluated using three standard automatic evaluation metrics: BLEU (Papineni et al., 2002), NIST (Doddington, 2002) and METEOR (Banerjee and Lavie, 2005).

5 Experiments

As described in the previous sections, we built four different configurations of the same system for two translation directions, French-to-English and English-to-French, and carried out translation

experiments. We considered the relative merits to PB-SMT of using data of which the source part actually corresponds to the original SL, meaning that the original translation direction and the translation direction to handle are consistent, *vs.* data where this condition is partially met or not met at all. We also considered the extent to which these relative merits depend on whether the translation direction is French-to-English or English-to-French.

For each translation direction, the evaluation of the different configurations was carried out in three different ways:

- in the first place, each configuration was evaluated against one 500-sentence testset selected according to the same criterion as to the original SL as the data it was trained on; therefore, the four testsets used at this stage are different from one another;
- then, each configuration is evaluated against each of the other three testsets; in other words, each configuration is evaluated against testsets where there is no or little overlap in terms of the original SL with the data it was trained on;
- finally, each configuration is evaluated against the unique 2000-sentence testset resulting from the union of all four individual testsets.

6 Results

In the following sections, we present the results and discuss the associated trends first for French-to-English and then for English-to-French. The highest scores are highlighted in bold; the lowest scores are in italics.

6.1 French-to-English

6.1.1 Individual evaluation

The translation quality of each configuration is measured individually against each 500-sentence testset. First, we give the scores (BLEU, NIST and METEOR) which each configuration achieves on its specific testset (Table 5), *i.e.* the testset which meets the same requirements as to the original SL; for instance config-1 is evaluated against test-1, config-2 against test-2, etc.

The results are consistent across all metrics. If we look for example at BLEU, we see a considerable absolute improvement of 0.0956 when moving from config-2, which achieves the lowest score (0.2008), to config-4, which performs best with a score of 0.2964. This might be due to the fact that for config-2 the French and English parts of the data bear less resemblance to each other. Both languages being translated from several other languages, they may present a higher proportion of divergences than if translated directly from one into another, thus making generalisation over the data less efficient. The second best configuration (0.2857) is config-3, *i.e.* the configuration which was trained on a corpus representing the reverse original translation direction, *i.e.* English-to-French. The third best (0.2608) is config-1 which uses data based on various original SL, thus including original French and English as well as translated French and English. Therefore, we conclude that data containing original French and English translated from French is optimal when building a system translating from French into English. Conversely, data comprising exclusively French and English translated from several other languages appears to be suboptimal.[4]

system	BLEU	NIST	METEOR
config-1	0.2608	5.9771	0.5758
config-2	*0.2008*	*5.1531*	*0.4867*
config-3	0.2857	6.4717	0.6082
config-4	**0.2964**	**6.5502**	**0.6162**

Table 5: French-to-English – evaluation on individual 500-sentence testsets

We further analyse how each configuration performs on each individual testset (Table 6). Here again the results are consistent across all metrics, and hence we present the results as measured by only one of the three metrics used in our experiments, BLEU.

system	test-1	test-2	test-3	test-4
config-1	0.2608	*0.2014*	0.2632	**0.2887**
config-2	0.2449	*0.2008*	0.2529	**0.2764**
config-3	0.2519	*0.1991*	**0.2857**	0.2695
config-4	0.2465	*0.1963*	0.2579	**0.2964**

Table 6: French-to-English – evaluation on all four individual 500-sentence testsets (BLEU)

[4]The results obtained for French-to-English by each configuration on its individual testset when MERT is performed confirm the observations made so far. Tests with MERT are currently ongoing for the experiments presented in the remainder of the paper.

We observe that config-3 and config-4 perform best on the testset which presents the same characteristics as the training data in terms of original SL: English as original SL for config-3/test-3 and French as original SL for config-4/test-4. We also note that both config-1 and config-2 achieve the best scores on test-4 rather than on the testsets that present the same characteristics as the training data in terms of the original SL, test-1 and test-2 respectively. On the other hand, all configurations achieve the lowest translation quality when it comes to translating test-2, which contains exclusively non-original French, *i.e.* French translated from languages other than English. A potential explanation for the latter observation may again lie in the resemblance between the source language being translated and the reference. It is probable that the references associated with test-4 bear a higher resemblance/are more faithful to the source since they were originally translated from French, whereas the opposite might be true for the references associated with test-1 and test-2 since only part or none of them was originally translated from French.

6.1.2 Overall evaluation

This time, each configuration is evaluated against the unique 2000-sentence testset resulting from the union of the individual testsets according to the same metrics as used previously (Table 7).

system	BLEU	NIST	METEOR
config-1-2000	**0.2542**	6.4797	0.5646
config-2-2000	*0.2424*	*6.3211*	*0.5525*
config-3-2000	0.2520	**6.5385**	0.5558
config-4-2000	0.2500	6.4331	**0.5681**

Table 7: French-to-English – evaluation on the unique 2000-sentence testset

First of all, we observe that the scores are lower when measured on the 2000-sentence testset in comparison with the individual 500-sentence testsets, for instance 0.2542 *vs.* 0.2964 for the best BLEU score. Moreover, the metrics give conflicting results. Only one score is consistent across all metrics on the one hand, and with the individual evaluations on the other hand: config-2 yields the lowest translation quality, *i.e.* 0.2424 BLEU. This confirms our previous conclusion: using data where both French and English are translated from other languages has a negative effect on MT performance and constitutes the least optimal training configuration.

Looking at the other scores, we can see that if we ignore NIST, then config-1 outperforms config-3. If we ignore METEOR, then config-3 outperforms config-4. There is a trend towards config-1 and config-3 being the best two configurations when translation is performed on a testset that mixes original French and French translated from English as well as other languages. In this respect, going back to Table 6, the following detailed observations can be drawn:

test-1: config-1>config-3>config-4>config-2
test-2: config-1>config-2>config-3>config-4
test-3: config-3>config-1>config-4>config-2
test-4: config-4>config-1>config-3>config-2

Config-1 outperforms config-3 on 3 out of 4 testsets. Config-3 outperforms config-4 on 3 out of 4 testsets. In at least one case — config-1 — the optimal results are obtained when there is an overlap in the contents of the training data and the testset in terms of original SL.

6.2 English-to-French

6.2.1 Individual evaluation

We now look at the opposite translation direction, *i.e.* English-to-French. The results are presented in Table 8. This time, config-3 is the one which matches the current translation direction since it is based on French translated from English and original English. To confirm the conclusions for French-to-English, config-3 should perform best.

system	BLEU	NIST	METEOR
config-1	0.2615	5.9315	0.5624
config-2	*0.1969*	*4.9954*	*0.4777*
config-3	0.2965	6.3787	0.5910
config-4	**0.3201**	**6.7205**	**0.6161**

Table 8: English-to-French – evaluation on individual 500-sentence testsets

As for French-to-English, scores are consistent across all evaluation metrics. Unexpectedly, the relative ranking turns out to be exactly the same as for French-to-English. Config-4 yields the highest translation quality (0.3201 BLEU) although in this case training was performed on a corpus the content of which represents the reverse translation direction with respect to the tested translation direction, meaning that the English part consists of

texts translated from French which is thus the original SL. Config-3 is second best. As previously, config-2 achieves the lowest score, *i.e.* 0.1969 BLEU. According to BLEU, there is an absolute increase of 0.1232 in performance when moving from config-2 to config-4, which corresponds to 38% relative increase. We also note that English-to-French translation yields better overall results than French-to-English on the same testset, 0.3201 BLEU *vs.* 0.2964 BLEU, which is unusual.

The performance of each configuration on each individual testset is shown in Table 9. The situation is similar as for French-to-English. Here again, config-3 and config-4 perform best on the testset which presents the same characteristics as the training data in terms of the original SL, whereas config-1 and config-2 yield the highest results on test-3 which contains original English. As previously, the lowest translation quality is obtained when translating test-2, which contains only English translated from other languages than French. Therefore, the results for English-to-French confirm the findings for the opposite translation direction.

system	test-1	test-2	test-3	test-4
config-1	0.2615	*0.1970*	**0.2814**	0.2661
config-2	0.2523	*0.1969*	**0.2731**	0.2602
config-3	0.2514	*0.1971*	**0.2965**	0.2649
config-4	0.2478	*0.2011*	0.2754	**0.3201**

Table 9: English-to-French – evaluation on all four individual 500-sentence testsets (BLEU)

6.2.2 Overall evaluation

Table 10 shows evaluation results on the 2000-sentence testset for English-to-French.

system	BLEU	NIST	METEOR
config-1	0.2517	6.3192	0.5478
config-2	*0.2459*	*6.2242*	*0.5406*
config-3	0.2525	6.3335	**0.5576**
config-4	**0.2616**	**6.4384**	0.5511

Table 10: English-to-French – evaluation on the unique 2000-sentence testset

Part of the observations we can make when looking at this table are similar to those made for the French-to-English experiments: translation quality is generally reduced compared to the evaluations made on the individual 500-sentence testsets, 0.2616 *vs.* 0.3201 BLEU score. Furthermore, the metrics give conflicting results; config-2 gives the lowest translation quality, *i.e.* 0.2459 BLEU, which is the only consistent result as far as all metrics and individual evaluations are concerned.

Looking at the other scores in Table 10, a different situation to that observed for the French-to-English direction arises. This time, if we ignore METEOR, config-4 outperforms config-3, config-3 outperforms config-1 and config-1 outperforms config-2. In other words, the tendency observed on the 2000-sentence testset is consistent with the scores measured on the individual testsets. This is quite unexpected: better translation quality is achieved although there is no overlap between the training corpus and the testset in terms of original SL. Furthermore, the contents of the training corpus were originally issued in French and translated into English, meaning that they represent the reverse translation direction with respect to the tested translation direction. We see that the detailed results are less clear-cut (more mixed) than for French-to-English upon looking at Table 9. Config-4 outperforms config-3 on 2 testsets out of 4; config-3 outperforms config-1 on 2 testsets out of 4.

7 Conclusions and Future Work

In this paper, we argued that the nature of the original SL should not be neglected as far as bilingual data for PB-SMT training is concerned. We observed that the original SL has a considerable impact on French–English PB-SMT training. First of all, using data where neither French nor English is the original SL, *i.e.* both are translated from several other languages, resulted in a clear-cut absolute decrease in translation quality in all scores, for instance up to 0.1232 in BLEU, and regardless of the translation direction considered. For French-to-English, evaluations on individual testsets showed that using data which contains as original SL the source language being translated proved to be the optimal configuration, leading to up to 0.0956 absolute increase in BLEU. However, overall evaluations on one unique testset indicated a tendency towards preferring data based on various original SLs.

System developers have not paid any attention to date to the role of the human translator in developing bilingual corpora for use as training data in PB-SMT. Our results demonstrate quite clearly

that this attitude has to change. Our findings are especially poignant to those whose mantra is "More data is better data" (cf. (Zollmann et al., 2008)), as again it is clear that what we *really* need is *better quality* data. In order to show more significant improvements in our PB-SMT systems, it appears that we might be better off paying translators to develop language pair-specific material for use as training data. Far from ever being made redundant by SMT systems, the role of the translator is even more crucial than has been acknowledged heretofore, and only closer relations between human translators and system designers are likely to lead to further improvements in translation quality in PB-SMT.

We are replicating the experiments with MERT and plan to work with a fixed language model. We will also scale up our experiments in order to investigate to what extent the observed trends are influenced by the amount of data. We will address two additional questions. Once all direct translations have been used, does it hurt to add data that was indirectly translated via another language? Given a full corpus, is it possible to improve translation quality by filtering out parts corresponding to indirect translations? Finally, we will run tests with different language pairs, particularly with languages from different families, and with different corpora provided that enough data is available.

Acknowledgements

We are grateful to Science Foundation Ireland (`http://www.sfi.ie`) grant 05/IN/1732 for funding this work.

References

Banerjee, S. and A. Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, 43th Annual Meeting of the Association of Computational Linguistics (ACL-05)*, Ann Arbor, MI, 65–72.

Bowker, L. 2003. Investigate 'reversible' translation resources: are they equally useful in both translation directions? *Speaking in Tongues: Language across Contexts and Users*, Luis Pérez Gonzáles ed. 201–224.

Brown, P. F., J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1993. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

Doddington, G. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. *Human Language Technology: Notebook Proceedings*, San Diego, CA, 128–132.

Gale, W. J., and K. W. Church. 1993. A Program for Aligning Sentences in Parallel Corpora. *Computational Linguistics*, 19(3):75–102.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *MT Summit X: The Tenth Machine Translation Summit*, Phuket, Thailand, 79–86.

Koehn, P., H. Hoang, A. Birch, Ch. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, Prague, Czech Republic, 177–180.

Koehn, P., F. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL'03)*, Edmonton, Canada, 48–54.

Och, F., and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA, 11–318.

Steinberger, R., B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufiş, and D. Varga. 2006. The JRC-Acquis: A multilingual Aligned Parallel Corpus with 20+ Languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, 2142–2147.

Stolcke, A. 2002. SRILM: an Extensible Language Modeling Toolkit. *Proceedings of the International Conference on Spoken Language Processing*, Denver, CO, 901–904.

Teubert, W. 1996. Comparable or Parallel Corpora? *International Journal of Lexicography*, 9(3):239–264.

Wu, H., and H. Wang. 2007. Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3):165–181.

Zollmann A., A. Venugopal, F. Och, and J. Ponte. 2008. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. In *Coling 2008, The 22nd International Conference on Computational Linguistics, Proceedings*, Manchester, UK, 1145–1152.

Word- and Sentence-level Confidence Measures for Machine Translation

Sylvain Raybaud, Caroline Lavecchia, David Langlois, Kamel Smaïli
PAROLE team, LORIA
Campus Scientifique BP 239
54506 Vandoeuvre-lès-Nancy FRANCE
{sylvain.raybaud,caroline.lavecchia,david.langlois,kamel.smaili}@loria.fr

Abstract

A machine translated sentence is seldom completely correct. Confidence measures are designed to detect incorrect words, phrases or sentences, or to provide an estimation of the probability of correctness. In this article we describe several word- and sentence-level confidence measures relying on different features: mutual information between words, n-gram and backward n-gram language models, and linguistic features. We also try different combination of these measures. Their accuracy is evaluated on a classification task. We achieve 17% error-rate (0.84 f-measure) on word-level and 31% error-rate (0.71 f-measure) on sentence-level.

1 Introduction

Statistical techniques have been widely used and remarkably successful in automatic speech recognition, machine translation and in natural language processing over the last two decades. This success is due to the fact that this approach is language independent and requires no prior knowledge, only large enough text corpora to estimate probability densities on. However statistical methods suffer from an intrinsic drawback: they only produce the result which is most likely given training and input data. It is easy to see that this will sometimes not be optimal with regard to human expectations. It is therefore important to be able to automatically evaluate the quality of the result: this can be handled by the different *confidence measures (CMs)* which have been proposed for machine translation.

This paper extends and improve the work presented in (Raybaud et al., 2009): we introduce new CMs to assess the reliability of translation results. The proposed CMs take advantage of the constituents of a translated sentence: n-grams, word triggers, and also word features. We also combine the scores given by the different measures in order to produce a new one, hopefully more powerful, and the scores given to the different words in order to estimate the whole sentence's reliability.

1.1 A brief overview of statistical machine translation

In this framework the translation process is essentially the search for the most probable sentence in the target language given a sentence in the source language; let $\mathbf{s} = s_1, .., s_I$ be the source sentence (to be translated) and $\hat{\mathbf{t}} = t_1, .., t_J$ be the sentence generated in the target language by the system:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{t}|\mathbf{s}) \quad (1)$$

which is equivalent (using the Bayes rule) to:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{t})P(\mathbf{s}|\mathbf{t}) \quad (2)$$

In Equation 2, $P(\mathbf{t})$ is estimated from a *language model* and is supposed to estimate the correctness of the sentence ("is it a good sentence in the target language ?"), and $P(\mathbf{s}|\mathbf{t})$ is computed from a *translation model* and is supposed to reflect the accuracy of the translation ("does the generated sentence carry exactly the same information than the source sentence ?"). The language model is itself estimated on a large text corpus written in the target language, while the translation model is computed on a bilingual aligned corpus (a text and its translation with line-wise correspondence).

Proceedings of the 13th Annual Conference of the EAMT, pages 104–111,
Barcelona, May 2009

The decoder then generates the best hypothesis by making a compromise between these two probabilities.
Of course there are three main drawbacks to this approach: first the search space is so huge that exact computation of the optimum is intractable; second, even if it was, statistical models have inherent limitations which prevent them from being completely sound linguistically; finally, the probability distribution P can only be estimated on finite corpora, and therefore suffers from imprecision and data sparsity. Because of that, any SMT system sometimes produces erroneous translations. It is an important task to detect and possibly correct these mistakes, and this could be handled by confidence measures.

2 An Introduction to Confidence Measures

2.1 Motivation and principle of confidence estimation

As said before, SMT systems make mistakes. A word's translation can be wrong, misplaced, or missing. Extra words can be inserted. A whole sentence can be wrong or only parts of it. In order to improve the overall quality of the system, it is important to detect these errors by assigning a so called confidence measure to each translated word, phrase or sentence. Ideally this measure would be the probability of correctness. An ideal word-level estimator would therefore be the probability that a given word appearing at a given position in a given sentence is correct; using the notations of Section 1.1 (t_j being the j-th word of sentence $\mathbf{t}$), this is expressed by the following formula:

$$\text{word confidence} = P(correct|j,t_j,\mathbf{s}) \quad (3)$$

and an ideal sentence-level estimator would be:

$$\text{sentence confidence} = P(correct|\mathbf{t},\mathbf{s}) \quad (4)$$

However these probabilities are difficult to estimate accurately; this is why existing approaches rely on approximating them or on computing scores which are supposed to monotonically depend on them.

2.2 State of the art

Confidence estimation is a common problem in artificial intelligence and information extraction in general (Culotta and McCallum, 2004; Gandrabur et al., 2006). When it comes to natural language processing, it has been intensively studied for automatic speech recognition (Mauclair, 2006; Razik, 2007; Guo et al., 2004). We find in literature (Blatz et al., 2003; Ueffing and Ney, 2004; Ueffing and Ney, 2005; Uhrik and Ward, 1997; Duchateau et al., 2002) different ways of approximating the probability of correctness or of calculating scores which are supposed to reflect this probability.

There exist three dominating approaches to estimation of word- and sentence-level confidence measures for machine translation:

- Estimate posterior probabilities (for example using a word-lattice or a translation table).
- Compute a predictive parameter (numerical score, for example a likelihood ratio) supposed to depend monotonically on the correctness probability.
- Combine predictive parameters through machine learning techniques in order to estimate the probability of correctness.

Many different confidence measures are investigated in (Blatz et al., 2003). They are based on source and target language models features, n-best lists, words-lattices, translation tables, and so on. The authors also present efficient ways of classifying words or sentences as "correct" or "incorrect" by using naïve Bayes, single- or multi-layer perceptron.

2.3 Our approach to confidence estimation

In the following we will first present three original word-level predictive parameters, based on:

- Intra-language mutual information (intra-MI) between words in the generated sentence.
- Inter-language mutual information (inter-MI) between source and target words.
- A target language model based on linguistic features.

We also implement two classical predictive parameters and combine them with our estimators:

- An n-gram model of the target language.
- A backward n-gram language model (Duchateau et al., 2002).

Mutual Information has been proved suitable for building translation tables (Lavecchia et al., 2007). We use intra-language MI to estimate the relevance of a word in the candidate translation given its context (it is supposed to reflect the lexical consistency). Inter-language MI based confidence estimation gives an indication of the relevance of a translation by checking that each word in the hypothesis can indeed be the translation of a word in the source sentence. N-gram, backward n-gram and linguistic features models estimate the lexical and grammatical correctness of the hypothesis. These different measures are then combined, either linearly with weights optimised with regard to error rate, or through logistic regression (Section 6). Each of these estimators produces a score for every word. This score is then compared to a threshold and the word is labelled as "correct" if its score is greater, or "incorrect" otherwise. This classification is then compared to a man made reference which gives an estimation of the efficiency of the measures, in terms of error rate, ROC curve and F-measure (Section 2.3.1). Finally we combine the word-level scores in order to compute sentence-level confidence measures. Each sentence is then classified as correct or incorrect by comparing its score to a threshold, and this decision is compared to a man-made decision in order to estimate the accuracy of the measure.

2.3.1 Evaluation of the confidence measures

As explained before, the CMs are evaluated on a classification task. We split the test corpus of our machine translation system into a development corpus (300 pairs of sentences) and a test corpus (200 sentences) for our confidence measures. We manually classified as correct or incorrect the words and sentences from these 500 French translation generated by Pharaoh (Koehn, 2004). Human were given few constraints; the first and most important one was "the first impression is the best"; the second one was "if a word makes no sense in the sentence or is really misplaced then it is wrong"; the third one was "a translation that does not contain essential information stated in the source sentence is wrong"; the last and most important one was "the first impression is the best".
We then ran our classifiers on the same sentences. A word was classified as correct if its score was above a given threshold. The results were then compared to the human-made references. We used the following metrics to estimate how well our classifier behaved; "item" refers either to "word" or "sentence":

Classification Error Rate (CER) is the proportion of errors in classification:

$$\frac{\text{number of incorrectly classified items}}{\text{total number of items}}$$

Correct Acceptance Rate (CAR or Sensitivity) is the proportion of correct items retrieved:

$$\frac{\text{number of correctly accepted items}}{\text{total number of correct items}}$$

Correct Rejection Rate (CRR or Specificity) is the proportion of incorrect items retrieved:

$$\frac{\text{number of correctly rejected items}}{\text{total number of incorrect items}}$$

F-measure is the harmonic mean of CAR and CRR:

$$F = \frac{2 \times CAR \times CRR}{CAR + CRR}$$

These metrics are fairly common in machine learning. Basically a relaxed classifier has a high CAR (most correct words are labelled as such) and low CRR (many incorrect words are not detected), while a harsh one has a high CRR (an erroneous word is often detected) and a low CAR (many correct words are rejected).
As the acceptance threshold increases, CAR decreases and CRR increases. The plot of *CRR vs. CAR* is called the ***ROC curve*** (*Receiver Operating Characteristic*). The ROC curve of a perfect classifier would go through the point (1,1), while that of the most naive classifier (based on random scores) is the segment joining (0,1) and (1,0). The ROC curve can therefore be used to quickly visualise the quality of the classifier: the higher above this segment a curve is, the better. We also plotted on the same diagrams F-measure and CER against CAR.

3 Software and Material Description

Experiments were run using an English to French phrase-based translation system. We trained a system corresponding to the baseline described in the *ACL workshop on statistical machine translation* (Koehn, 2005). It uses an IBM-5 model (Brown et al., 1994) and has been trained on the EUROPARL corpus (proceedings of the European Parliament, (Koehn, 2005)) using GIZA++ (Och and Ney,

2000) and the SRILM toolkit (Stolcke, 2002). The decoding process is handled by Pharaoh. The French vocabulary was composed of 63,508 words and the English one of 48,441 words. We summarise in Table 1 the sizes of the different parts of the corpus. This system achieves state of the art performances.

set	sentences pairs	running words	
		English	French
Learning	465,750	9,411,835	10,211,388
Development	3000	75,964	82,820
Test	500	4,945	4,899

Table 1: Corpora sizes

Human annotators reported 16.5% erroneous words and 32.6% erroneous sentences, according to the previously stated criteria.

4 Mutual Information based Confidence Measures

4.1 Mutual information in language modelling

In probability theory mutual information measures how mutually dependent are two random variables. It can be used to detect pairs of words which tend to appear together in sentences. Guo proposes in (Guo et al., 2004) a word-level confidence estimation for speech recognition based on mutual information. In this paper we will compute inter-word mutual information following the approach in (Lavecchia et al., 2007), which has been proved suitable for generating translation tables, rather than Guo's.

$$MI(x,y) = p(x,y)log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) \quad (5)$$
$$p(x,y) = \frac{N(x,y)}{N}$$
$$p(x) = \frac{N(x)}{N}$$

where N is the total number of sentences, $N(x)$ is the number of sentences in which x appears and $N(x,y)$ is the number of sentences in which x and y co-occur. We smooth the estimated probability distribution, as in Guo's paper, in order to avoid null probabilities:

$$N(x,y) \leftarrow N(x,y)+C \quad (6)$$
$$p(x,y) \leftarrow \frac{p(x,y)+\alpha p(x)p(y)}{1+\alpha} \quad (7)$$

in which C is a non-negative integer and α a non-negative real number. For example, words like *"ask"* and *"question"* have a high mutual information, while words coming from distinct lexical fields (like *"poetry"* and *"economic"*) would have a very low one. Since it is not possible to store a full matrix in memory, only the most dependent word pairs are kept: we obtain a so called *triggers list*.

4.2 Confidence measure based on intra-language mutual information

By estimating which target words are likely to appear together in the same sentence, intra-language MI based confidence score is supposed to reflect the lexical consistency of the generated sentence. The source sentence is not taken into account. We computed mutual information between French words from the French part of the bilingual corpus. Table 2 shows an example of French intra-lingual triggers, sorted by decreasing mutual information.

word	→	*triggered word*
sécurité	→	*alimentaire*
sécurité	→	*étrangère*
sécurité	→	*politique*
	...	
politique	→	*commune*
politique	→	*économique*
politique	→	*étrangère*

Table 2: An example of French intra-lingual triggers

Let $\mathbf{t} = t_1..t_J$ be the generated sentence. The score assigned to t_j is the weighted average mutual information between t_j and the words in its context:

$$C(t_j) = \frac{\sum_{i=1..J, i\neq j} w(|j-i|)MI(t_i,t_j)}{\sum_{i=1..J, i\neq J} w(|j-i|)} \quad (8)$$

where $w()$ is a scaling function lowering the importance of long range dependencies. It can be constant if we do not want to take words' positions into account, exponentially decreasing if we want to give more importance to pairs of words close to each other, or a shifted Heaviside function if we want to allow triggering only within a given range (which we will refer to as *triggering window*). Function words (like *"the"*, *"of"*,...) generally have a very high mutual information with all other words thus polluting the trigger list; therefore they are not taken into account for computing mutual

information.

Presenting the performances of the confidence measure with all different settings (different triggering windows, size of trigger list,...) would be tedious. Therefore we only show the settings that yield the best performances. Note that while other settings often yield much worse performance, a few perform almost as well, therefore there are no definite "optimal settings". Figure 1 shows the ROC curve, CER and F-measure of a classifier based on intra-MI in which function words were ignored.

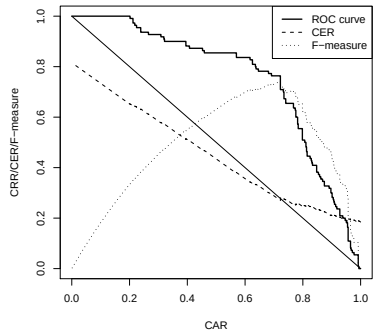


Figure 1: Intra-MI, no function words, no weighting nor triggering window.

Taking word positions into account yields lower performance: intra-language MI indeed reflects lexical consistency of the sentence, but two related words may not be next to each other in the sentence.

4.3 Confidence measure based on inter-language mutual information

The principle of intra-language MI was to detect which words trigger the appearance of another word in the same sentence. This principle can be extended to pairs of source and target sentences (Lavecchia et al., 2007): let $N_S(x)$ be the number of source sentences in which x appears, $N_T(y)$ the number of target sentences in which y appears, $N(x,y)$ the number of pairs *(source sentence, target sentence)* such that x appears in the source and y in the target, and N the total number of pairs of source and target sentences. Then let us define:

$$\begin{aligned} p_S(x) &= \frac{N_S(x)}{N} \\ p_T(y) &= \frac{N_T(y)}{N} \\ p(x,y) &= \frac{N(x,y)}{N} \\ MI(x,y) &= p(x,y)log_2\left(\frac{p(x,y)}{p_S(x)p_T(y)}\right) \end{aligned} \quad (9)$$

Guo's smoothing can be applied as in Section 4.2. One then keeps only the best triggers and obtain a so-called *inter-lingual triggers list*. Table 3 shows an example of such triggers between English and French words, sorted by decreasing mutual information.

English word	→	*triggered French word*
security	→	*sécurité*
security	→	*étrangère*
security	→	*politique*
	...	
policy	→	*politique*
policy	→	*commune*
policy	→	*étrangère*

Table 3: An Example of Inter-Lingual triggers

The confidence measure is then:

$$C(t_j) = \frac{\sum_{i=1}^{I} w(|j-i|)MI(s_i,t_j)}{\sum_{i=1}^{I} w(|j-i|)} \quad (10)$$

We show in Figure 2 the characteristics of such an inter-MI based classifiers. This time triggering was allowed within a window of width 9 centred on the word the confidence of which was being evaluated. Function words were excluded.

Unlike intra-MI based classifier, we found here that setting a triggering window yields the best performance. This is because inter-language MI indicates which target words are possible translations of a source word. This is much stronger than the lexical relationship indicated by intra-MI; therefore allowing triggering only within a given window or simply giving less weight to "distant" words pairs reflects the fact that words in the source sentence and their translations in the target sentence appear more or less in the same order (this is the same as limiting the distortion, which is the difference between the positions of a word and its translation).

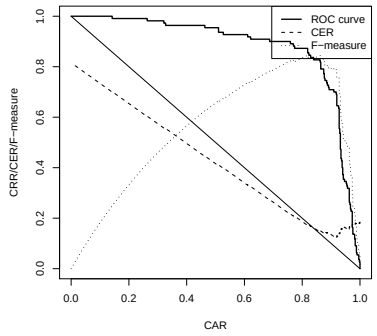


Figure 2: Inter-language MI based CM: function words excluded, no normalisation, triggering is allowed within a centred window of width 9.

5 Language-Model based Confidence Measures

We now present confidence measures based on different n-gram-like target-language models. We assume that if a sentence "looks wrong" in the target language then it is unlikely to be an accurate translation. We will not present their word-level performance, which are somewhat poor, however we will see in Section 7 that they are efficient for detecting incorrect sentences.

5.1 N-grams based confidence measure

Remember Equation 2: the decoder makes a compromise between $P(\mathbf{t})$ (which we will refer to as *language model score*) and $P(\mathbf{s}|\mathbf{t})$ (*translation score*). Because of that, if a candidate $\mathbf{t}$ has a high translation score and a low language model score, it might be accepted as the "best" translation. But a low LM score often means an incorrect sentence and therefore a bad translation. This consideration applies on sub-sentence level as well as on sentence level: if the n-gram probability of a word is low, it often means that it is wrong or at least misplaced. Therefore we want to use the language model alone in order to detect incorrect words. We decided to use the word probability derived from an n-gram model as a confidence measure:

$$C(t_j) \quad = \quad P(t_j|t_{j-1},..,t_{j-n+1}) \qquad (11)$$

While intra-language triggers are designed to estimate the lexical consistency of the sentence, this measure is supposed to estimate its well-formedness. We empirically found that 4-grams were best suited.

5.2 Backward n-gram language model

Because classical n-gram models only take into account the left context of a word, it is natural to extend the idea to consider the *right-context* (Duchateau et al., 2002). This should be efficient to detect, for example, incorrect determinants and other function words. A backward n-gram language model is simply trained on a corpus in which sentences have been "reverted": "*Hello world !*" becomes "*! world Hello*". We then use as a confidence measure:

$$C(t_j) \quad = \quad P(t_j|t_{j+1},..,t_{j+n-1}) \qquad (12)$$

We found that bigrams achieved the best performances, which backs our idea that this language model is useful for detecting wrong function words.

5.3 Linguistic features based confidence measure

We designed a confidence measure to specifically target grammatical errors. using BDLEX (De Calmès and Pérennou, 1998), each word $t$in the corpora was replaced by a vector $\tilde{t}$ of it's *syntactic class*, *tense* if relevant, and *number and gender* or *person*. We then built n-gram models on the modified training corpus, and used $P(\tilde{t}_j|\tilde{t}_{j-1},..,\tilde{t}_{j-n+1})$ as a confidence score. The performance were poor both at word- and sentence-level, therefore this measure won't be used in the rest of the paper. More information can be found in (Raybaud et al., 2009).

6 Fusion of Confidence Measures

We linearly combined the scores assigned to each word by different confidence measures to produce a new score. The weights are optimised with respect to error-rate on our development corpus. This method yields no significant improvement on the best measure used alone (inter-language mutual information 4.3). Therefore we used a more sophisticated logistic regression instead.

6.1 Logistic regression

An other option is to use logistic regression to estimate a probability of correctness given a vector of predictive parameters. If $X \in \mathbb{R}^k$ is a vector of predictive parameters, the idea of logistic regression is to find coefficients $\Theta \in \mathbb{R}^k, b \in \mathbb{R}$ such that:

$$P(correct|X) = \frac{1}{1+e^{\langle\Theta,X\rangle+b}} \qquad (13)$$

These coefficients are optimised with respect to the maximum likelihood criterion. Here again we could not improve word-level performances compared to inter-language mutual information; the latter is way better than any other measure we implemented, thus being difficult to improve on. however we will see that this performed well on sentence-level.

7 Sentence-level Confidence Estimation

We chose to estimate a sentence's reliability from the confidence score of its words. We empirically found that the best method was to combine LM and backward LM confidence measures through logistic regression, and then set the sentence's score as the normalised product of the correctness probabilities of words; let $X(t) \in \mathbb{R}^2$ be a vector whose components are LM and backward-LM probabilities of word t; let $\Theta \in \mathbb{R}^2$ and $b \in \mathbb{R}$ be the optimal logistic regression coefficients; then the score of sentence $\mathbf{t} = t_1, .., t_J$ is given by:

$$P(correct|j,\mathbf{t}) = \frac{1}{1+exp\left(\langle\Theta, X(t_j)\rangle + b\right)} \quad (14)$$

$$C(\mathbf{t}) = \sqrt[J]{\prod_{j=1}^{J} P(correct|j,\mathbf{t})} \quad (15)$$

Figure 3 shows the ROC curve, f-measure and error rate curves of a sentence classifier relying on the above combination of measures. The best f-measure is 0.71, corresponding to a 30.6% error rate.

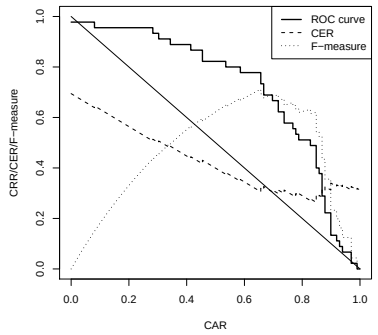


Figure 3: Detection of incorrect sentences based on n-gram and backward-n-gram language models.

8 Discussion and Conclusion

In this article, we present confidence scores that showed interesting discriminating power. We summarised the results obtained by the best different word-level estimators (in terms of F-measure) in Table 4. For comparison Blatz et al. obtain in (Blatz et al., 2003) a CER of 29.2% by combining two different word posterior probability estimates (with and without alignment) and the translation probabilities from IBM-1 model. The result obtained with sentence classifiers are presented in Table 5. For comparison Blatz et al. obtained an error rate around 28%.

	CER	CAR	CRR	F-measure
intra-MI	0.270	0.722	0.764	0.742
inter-MI	0.171	0.819	0.873	0.845

Table 4: Performances of the best word-classifiers.

	CER	CAR	CRR	F-measure
LM and LM-backward	0.306	0.657	0.778	0.712

Table 5: Performances of the best sentence-classifier.

It is interesting to remark that the confidence measures which perform well at sentence level are those who perform poorly at word level. It might be because sometimes while you can tell for sure that a sentence is wrong, it is difficult to pinpoint an erroneous word. Also an important cause of sentence incorrectness is wrong word order, about which MI based confidence measures are lenient, while LM based ones are not.

8.1 Application of Confidence Measures

Beside manual correction of erroneous words we can imagine several applications of confidence estimation: **pruning or re-ranking of the n-best list**, **generation of new hypothesis** by recombining parts of different candidates having high scores, or **discriminative training** by tuning the parameters to optimise the separation between sentences (or words, or phrases) having a high confidence score (hopefully they are correct translations) and sentences having a low one.

8.2 Prospects

We plan to go further in our investigation on confidence measures for SMT: first the measures we used do not directly take into account word deletion nor word order, neither do our reference corpus (missing words are not indicated). This serious drawback has to be addressed. Also many features used in speech recognition or automatic translation could be used for confidence estimation: distant models, word alignment, word spotting, etc...

We also plan to investigate SVM and neural network for combining predictive parameters (Zhang and Rudnicky, 2001). Finally we have to work on the corpora themselves: man-made classification is slow, tedious, and the results depend heavily on the operator. We will investigate semi-automatic creation of labelled training, development and test data for confidence measures.

References

Blatz, J., E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing. 2003. Confidence estimation for machine translation. final report, jhu/clsp summer workshop.

Brown, P.F., S.D. Pietra, V.J.D. Pietra, and R.L. Mercer. 1994. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Culotta, A. and A. McCallum. 2004. Confidence estimation for information extraction. *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

De Calmès, M. and G. Pérennou. 1998. Bdlex: a lexicon for spoken and written french. In *Proceedings of 1st International Conference on Langage Resources & Evaluation*.

Duchateau, J., K. Demuynck, and P. Wambacq. 2002. Confidence scoring based on backward language models. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002.*, 1.

Gandrabur, S., G. Foster, and G. Lapalme. 2006. Confidence estimation for nlp applications. *ACM Transactions on Speech and Language Processing*, 3(3):1–29.

Guo, G., C. Huang, H. Jiang, and R.H. Wang. 2004. A comparative study on various confidence measures in large vocabulary speech recognition. *2004 International Symposium on Chinese Spoken Language Processing*, pages 9–12.

Koehn, P. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, pages 115–124.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. *MT Summit*, 5.

Lavecchia, C., K. Smaïli, D. Langlois, and J.P. Haton. 2007. Using inter-lingual triggers for machine translation. *Eighth conference INTERSPEECH*, pages 2829–2832.

Mauclair, J. 2006. *Mesures de confiance en traitement automatique de la parole et applications*. Ph.D. thesis, LIUM, Le Mans, France.

Och, F.J. and H. Ney. 2000. Giza++: Training of statistical translation models. *available at http://www.fjoch.com/GIZA++.html*.

Raybaud, S., C. Lavecchia, D. Langlois, and K. Smaïli. 2009. New confidence measures for statistical machine translation. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 61–68.

Razik, Joseph. 2007. *Mesures de Confiance trame-synchrones et locales en reconnaissance automatique de la parole*. Ph.D. thesis, LORIA, Nancy, FRANCE.

Stolcke, A. 2002. Srilm – an extensible language modeling toolkit. pages 901–904.

Ueffing, N. and H. Ney. 2004. Bayes decision rule and confidence measures for statistical machine translation. pages 70–81. Springer.

Ueffing, N. and H. Ney. 2005. Word-level confidence estimation for machine translation using phrase-based translation models. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 763–770.

Uhrik, C. and W. Ward. 1997. Confidence metrics based on n-gram language model backoff behaviors. In *Fifth European Conference on Speech Communication and Technology*, pages 2771–2774.

Zhang, R. and A.I. Rudnicky. 2001. Word level confidence annotation using combinations of features. In *Seventh European Conference on Speech Communication and Technology*, pages 2105–2108.

Translating Questions for Cross-Lingual QA

Jörg Tiedemann
Information Science
University of Groningen
PO Box 716
9700 AS Groningen, The Netherlands
j.tiedemann@rug.nl

Abstract

In this paper we investigate possibilities of the development of a task-specific translation component for cross-language question answering. We focus on the optimization of phrase-based SMT for models trained on very limited data resources. We also look at the combination of such systems with another approach based on example-based MT with proportional analogies. In our experiments we could improve a strong baseline of a general purpose MT engine with more than 5 BLEU points.

1 Introduction

Question answering (QA) is a popular task in the NLP research community. It combines various exciting sub-tasks coming from research in information extraction, retrieval and different kinds of linguistic processing in a real-world application. Cross-lingual QA adds yet another component to such systems, namely a translation component, in order to open QA systems for different languages. The motivation is to enable users to post questions in their favorite language and to make it possible to find answers in documents using other languages. So far, cross-lingual QA is mainly of academic interest because of the general shortcomings in the accuracy of QA systems and also the quality of current general-purpose machine translation (MT) engines. In this paper, we investigate the use of standard techniques in statistical MT (SMT) for the development of a task-specific translation component to be integrated in a cross-lingual QA application. However, we focus exclusively on this component without evaluating the effect of translation quality on a particular QA engine.

The predominant approach to cross-lingual QA is to translate incoming questions into the language the QA system understands and then to run the system as usual. The answers are usually not translated, which is, especially for factoid questions, often also not necessary. Furthermore, users may well be able to browse through answers in another language (or may use on-line translation services to get a general understanding) but still feel more comfortable in asking in their own language. In our study, we will also follow this approach and, therefore, concentrate on the translation of questions. In particular, we take the case of English-to-Dutch question answering, mainly because of our interest in Dutch QA.

The simplest approach to cross-lingual QA is to use available on-line MT engines for the translation of questions (see for instance Larosa et al. (2005)). There are several problems with this approach: First of all, most of these engines are general-purpose MT systems which are not optimized for the specific task of translating questions. However, questions have a very specific syntactic structure, often very different from other sentence types. They often show similar patterns, especially factoid questions, which makes them suitable for a data-driven approach modeling these specific patterns in particular.

Another problem with on-line services is their availability and reliability. A cross-lingual QA system using such services always depends on these external resources and has to adjust to service changes and quality differences. For example, we experimented with Google Translate and observed differences in its behavior from one day to another, which seriously affected our QA pro-

Proceedings of the 13th Annual Conference of the EAMT, pages 112–119,
Barcelona, May 2009

totype: Google started to treat names (unknown to the system) in an unpredictable way, for instance, translating the German city name "Wernigerode" into "Waterloo". This, of course, has severe consequences for a QA system trying to answer questions such as "Where is Wernigerode?" or "How many people live in Wernigerode?". Later, however, this behavior was corrected by Google.

To sum-up: in order to build a simple cross-lingual QA system one needs a proper translation component. In order to reduce the dependency on on-line services and, especially, in order to improve translation quality we like to develop a task-specific translation component for our system. In this study we investigate if we can use standard techniques for doing this. Especially we like to see how far we can get with extremely scarce resources when optimizing with linguistic features and additional resources such as term databases.

The paper is organized as follows: In the next section we present the general setup including a brief discussion of the baseline and the approaches applied. Thereafter, various experiments are presented, and, finally we summarize our findings with some conclusions.

2 General Setup

In our experiments, we focus on factoid questions, especially the ones used at the QA tracks at CLEF. This is mainly due to the availability of data for training and testing. In the following some more details of the data collected are given. Thereafter, we briefly summarize the baseline scores using Google Translate and and the MT approaches applied in the experiments below.

2.1 Data

Cross-lingual QA has been a shared task at CLEF for several years. There are various multilingual resources available via CLEF which we are grateful for. In particular, we use the Multi-eight-04 corpus, a collection of 700 questions in eight languages (Magnini et al., 2005), the DISEQuA corpus, a collection of Dutch, Italian, Spanish and English questions (Magnini et al., 2003a), and the Multi-six corpus, a collection of 200 questions in six languages collected from CLEF QA-2003 (Magnini et al., 2003b). Altogether, this amounts to 1349 questions with English and Dutch translations.

Additionally, we also have one source of Dutch questions coming from the popular Winkler-Prins game (a Dutch quiz game similar to Trivial Pursuit), which we have used previously for training our disambiguation module when parsing questions (Bouma et al., 2005). This monolingual corpus contains 4509 questions.

Another resource that we use is the multilingual Europarl corpus (Koehn, 2005) with its more than 1,000,000 parallel sentences. From this corpus we also extracted 31,506 questions by simply searching for lines ending with question marks in both, English and Dutch translations. Certainly, these are not the typical questions to be expected as input for a QA system. However, they are still useful as they represent the specific syntactic structures of questions.

Finally, we also collected multilingual term databases from Wikipedia and Geonames.org. From the latter, we simply extracted all pairs of Dutch and English place names giving us 55,381 entries. From Wikipedia we made use of the link structure between Dutch and English pages and extracted 145,510 pairs of Wikipedia lemmas.

2.2 Baselines

The baseline refers to the approach of applying available general purpose MT engines. We have chosen to use the popular service by Google (Goo, 2008). For evaluation purposes we took 100 questions from our parallel data which will be applied in all experiments below. We are aware that this test set is very small but we had to compromise due to the size of the material available to us. Table 1 shows the BLEU scores[1] obtained when translating our test set with Google (English to Dutch) and scoring with the one reference translation per question given in the data.

Google Translate	BLEU
October 2008	31.09
November 2008	32.66
January 2009	32.45

Table 1: Translating the test set of 100 English questions to Dutch (on three different dates).

The "Google" baseline can be seen as a very strong baseline as it is a running system with many satisfied users. Also, manually inspecting the translations show that the quality is reasonable

[1]All BLEU scores are computed using the `multi-bleu.perl` script from the Moses package.

and most of the translated questions are indeed correct or at least understandable.

In table 1 above we can see the general problem of on-line services as we have discussed earlier. The system is in development and its behavior is not stable. Although, the BLEU scores are not very different, the output can vary quite a lot. In the introduction we already mentioned the issue of wrongly translating place names at some point. In the version of October 2008 we observed another issue which is quite important for our QA system: Google Translate did not recognize several Wh-words correctly but translated them as relative pronouns. Consider the following examples:

English: Who is the Prime Minister of Ireland ?
Dutch: *Die* is de premier van Ierland ?

English: When was Elvis Presley' s first record recorded ?
Dutch: *Toen* was Elvis Presley's eerste record geregistreerd?

English: For which film did Robert Bresson win the Grand Prix at Cannes ?
Dutch: Voor *die* film deed Robert Bresson wint de Grand Prix in Cannes ?

This, of course, is a serious problem for a QA system that uses patterns involving Wh-words in its question analysis when looking for the question focus. However, this problem seems to be solved in later versions of the on-line engine.

2.3 PSMT for Question Translation

Phrase-based statistical machine translation (PSMT) is currently extremely popular and can be seen as one of the state-of-the-art approaches in today's machine translation research. Its popularity is also due to the availability of tools for building statistical models (word aligners and phrase extractors) and for the actual translation (decoders). The techniques are becoming so well-known that we omit the general introduction of the (P)SMT approach and just refer to standard literature (see for example (Brown et al., 1993; Och and Ney, 2003; Koehn et al., 2007; Koehn and Hoang, 2007)).

It might come as a surprise to see PSMT as one of the approaches applied here after the introduction of our training data. SMT usually requires large amounts of training data (for instance, more than 1,000,000 sentences of parallel data). However, for our task-specific approach we only have a tiny amount of translated questions available. On the other hand, we know that questions (especially factoid questions) follow very regular patterns. They are often very short and usually do not include embedded clauses or other complex structures. The general question we want to ask here is: Can a small amount of very regular, task-specific training data be used for training a statistical model that can compete with larger models? We also like to know how far we can get when adding additional resources and tweaking the system in such a way that it maximizes the performance possible with the data available. Finally, we also want to see the effect of domain/task-specific data when combined with out-of-domain data, also in comparison with our strong baseline.

In our experiments we apply the Moses system (Koehn et al., 2007) and its accompanying tools such as GIZA++ (Och and Ney, 2003) and IRSTLM (Frederico et al., 2008). We mainly use standard settings for all components if not stated otherwise.

2.4 EBMT using Proportional Analogies

The idea of example-base machine translation (EBMT) using proportional analogies has been introduced by Lepage and Denoual(2005). The idea is to solve string-level analogies in order to translate new sentences given a database of example translations. For this no pre-processing, sentence decomposition, word level alignment nor any other type of training or generalization is needed. The translation process entirely relies on solving analogical equations. Proportional analogies are denoted as $A : B :: C : D$, which is to be read as "A is to B as C is to D". An example of such an analogy is given in figure 1.

It walk s	:	It walk ed	::	It float s	:	It float ed
across the		across the		across the		across the
str e et .		str e et .		riv e r .		riv e r .

Figure 1: An example of a proportional analogy.

In a parallel corpus all sentences are aligned to corresponding translations in another language. Proportional analogies in the source language can now be used to identify existing entries in the corpus for sentences that are not part of the corpus. Corresponding analogical equations on the target language side can then be used to actually find the translation of these new incoming sentences. The following summarizes the translation process:

example database: $X = (X_{src}||X_{trg})$
input sentence: D (to be translated)

1. $\forall A_i, B_i \in X_{src}$ solve $A_i : B_i :: x : D$

2. $\forall x = C_{i,j}$ solve $\widehat{A_{ik}} : \widehat{B_{ik}} :: \widehat{C_{i,j}}^{k} : y$ where $(\left[S_i, \widehat{S_{i,k}}\right] \in X)$[2]

3. sort solutions $y = \widehat{D_{i,j}}^{k,l}$ by frequency[3]

recursion: possible after step 1: $\forall x = C_{i,j} \notin X_{src}$ translate them with the procedure above and add solutions to the corpus X

This approach has been successfully applied to various language pairs in a specific domain (travel and tourism). For more details we refer to the background literature (Lepage and Denoual, 2005).

There are several reasons why this approach is quite appealing for our task, the translation of questions. Firstly, it does not use a statistical model and, therefore, does not require huge amounts of training data with similar contents to get reliable counts. (However we still require a good amount of examples to obtain a reasonable coverage.) Secondly, we believe that this approach works best with rather short sentences for which reasonable analogies can be found in a limited amount of time. The questions we deal with are rather short and, therefore, seem to be appropriate. Thirdly, the parallel database can easily be extended by other translation data, for example, databases of translated terms. As we know, factoid questions often use very regular structural patterns. Simple analogies can be used to replace, for example, named entities that are part of the question focus. Consider the following, very simple example:

- input sentence: 'What is the capital of Armenia?'
- bitext:

What is the capital of Somalia?
Wat is de hoofdstad van Somalië?
Flag of Armenia
Vlag van Armenië
Flag of Somalia
Vlag van Somalië

Flag of Somalia	: What is the capital of Somalia?	::	**Flag of Armenia**	: What is the capital of Armenia?
Vlag van Somalië	: What is de hoofdstad van Somalië?	::	Vlag van Armenië	: **Wat is de hoofdstad van Armenië?**

For our experiments we will use our small set of example questions augmented with the term databases extracted from Geonames and from Wikipedia. However, considering the size of our example corpus of questions we do not expect to find many solutions using this approach but the ones found are expected to be highly accurate.

[2]Indeces j, k, l are added to indicate that there are several solutions possible when solving analogical equations.

[3]The same solution can be often be found in various ways with the procedure above. Frequency is assumed to be a good indicator for preference.

3 Experiments

We will now turn to the actual experiments. We will use the data as described in the previous section, in particular, we will use the same evaluation set of 100 questions for all experiments listed below. Furthermore, for training the PSMT models we will use a development set of 100 questions taken from the training data for tuning model parameters with minimum error rate training. In the following we will look at individual experiments. A summary of our results is shown in section 4.

3.1 Different Types of Training Data

In the first experiment we compare PSMT models trained on our tiny task-specific corpus with one trained on much larger material (namely data from the Europarl corpus). Table 2 shows the BLEU scores obtained after training and tuning with standard settings of the Moses system.

language model & translation model	BLEU
CLEF	26.60
CLEF+terms	28.53
EP	27.20

Table 2: BLEU scores for PSMT models trained on tiny task-specific data (CLEF) and on larger parallel training data (Europarl = *EP*); *terms* refers to Wikipedia lemmas and Geonames

In all settings we use the target language part of the parallel training data for estimating the language model probabilities. We can see that the tiny model almost performs as well as the one trained on much larger material according to BLEU scores. The tiny model suffers a lot from unknown words, among them many named entities. Adding Wikipedia lemmas and translated Geonames improves the system a lot and the performance even passes the larger model now.

In a second experiment we like to investigate the influence of the language model. In particular, we like to see how important is the use of appropriate data when building the language model. Table 3 summarizes our results.

	translation model		
language model	CLEF	+terms	EP
EPq	27.94	27.73	28.33
CLEF+EPq	28.59	30.30	30.49
CLEF+WP+EPq	28.91	30.36	31.10
CLEF+terms+WP+EPq	29.51	31.86	30.08

Table 3: Different language models for basic PSMT settings. *EPq* refers to questions from the Europarl corpus.

As we can see, the performance improves significantly when using language models consisting of questions only (except the terms added in the last setting). We argued already earlier that a system should make use of the specific syntactic patterns of questions and the results in table 3 demonstrate the success of adapting the language model (which is mainly responsible for grammaticality and fluency in the target language) to this kind of data. Observe that the language model using Europarl questions outperforms the one estimated on the entire Europarl corpus when combined with the translation model from the same corpus. Adding small amounts of task-specific data (CLEF) improves the scores even further.

Finally, we like to see the influence of task-specific training data for estimating both, translation model and language model, when combined with larger out-of-domain data. Table 4 shows the BLEU scores obtained for various data sets.

	translation model	
language model	CLEF+EP	CLEF+terms+EP
CLEF+EP	33.27	33.76
CLEF+EPq	36.34	35.21
CLEF+WP+EPq	36.79	34.76

Table 4: Combining task-specific and out-of-domain data.

The results show clearly that it is still helpful to add more data when building statistical MT models. However, in-domain data (even tiny amounts) are very important also for the translation model as we can see in the BLEU scores above. The results are all above the previous ones and now also exceed the Google baselines. Note that the term databases do not add anything to the model anymore when combining the CLEF data with the larger Europarl corpus. This probably means that the necessary terms are already included in the data and further databases are not necessary.

Finally, we want to mention that we also tried to use various combinations of in-domain and out-of-domain models (language models and phrase tables) as separate factors in the log-linear PSMT model and alternative paths during decoding. However, after minimum error rate tuning the scores were similar or below the ones presented above and, therefore, we omit these experiments.

3.2 Factored Models

One of the important extensions in the Moses system is the support of so-called factored translation models (Koehn and Hoang, 2007). It actually provides a framework for the integration of linguistic features or any other word-level features to be integrated in the translation models, the language models to be used in combination by the Moses decoder. There are many possibilities for an integration of such extra features. For example, a phrase table can combine surface word forms with POS tags and translate them into corresponding word roots with attached POS labels for the target language. Factors can also be translated separately using different phrase tables. They can even be generated on the target language side from other factors. In this way translation decisions can be based on various factors allowing different kinds of generalizations, sparseness of data can be reduced for example by the use of lemmas instead of word forms together with a target language generation step and fluency of the output can be improved by the integration of language models over different features.

In order to test various settings using factored models we parsed our data with Alpino (van Noord, 2006) on the Dutch side and extracted word-level features from the dependency graphs created by the parser. In this way we got the following factors: root forms, coarse POS tags, fine-grained POS tags with morphosyntactic information and dependency relations (to the corresponding head word). For English we used the C&C tools (Curran et al., 2007) to tag the data directly with POS tags and CCG supertags. After doing this we ran various experiments with different settings for factors and translation and generation steps. Unfortunately, the results so far are quite disappointing. We omit most of our results and just list a few of the better example in table 5 below.

Unfortunately, no significant and consistent im-

	translation model	
LM = CLEF+EPq	CLEF	CLEF+EP
baseline	28.59	36.34
w→ l,p+c→ p+r,l+p+r→ w	27.97	33.31
w→ l,p→ p,l+p→ w	29.16	30.14
w→ w, generate p	28.90	36.18

Table 5: Example settings of factored PSMT trained on CLEF and CLEF+EP (w=wordform, l=lemma, r=dependency relation to head, p=POS)

provements could be measured with the settings shown in the table. The first setting refers to a model with two translation steps (words to lemmas and POS-tags+CCG supertags to POS tags and dependency relations) and one generation step (lemmas+POS tags+dependency relations to surface word forms). The second setting refers to a model with also two translation steps (word to lemmas and POS tags to POS tags) and a generation step (lemma+POS to surface words). Unfortunately, the performance drops for all settings.

The last setting in table 5 refer to a standard approach (word to word translation) with a generation step added to generate POS tags. The reason for doing this is to add a POS language model into the decoding process for better generalization. However, no consistent improvement can be seen here.

Similar behavior could be observed for other kinds of factored models we have tried so far. In most cases we observed decreasing performances. More investigations are required to get a clear picture of the capability of factored translation models.

3.3 Escaping Named Entities

We already mentioned earlier that questions follow similar patterns and often differ only in certain named entities being part of the question focus. One idea is to escape the named entities from the statistical model and to translate them separately in a second step. Here again, we are interested in how far we can get with small amounts of training data. Replacing named entities with a dummy variable modifies the training material in such a way that these regular patterns should be more visible for a statistical approach and, thus, the model should become more general.

We used the following procedure: First we replaced named entities (NE) with a special dummy word and trained the PSMT models on the modified data. Here we used a very simplistic approach to detect NE's by replacing all (sequences of) capitalized words with the dummy word in source and target language. In the translation step we simply applied the models as usual and thereafter replaced dummy words in the output with name translations from our Wikipedia/Geonames database. Unknown names are simply copied as usual and if there are less variables in the output than in the input we added the names at the end of the translated question. This is certainly a very simplified procedure and only of conceptual interest. The results of applying this approach are shown in table 6.

	translation model	
language model	CLEF	CLEF+EP
CLEF	30.40 (32.80)	28.37 (29.85)
CLEF+EPq	30.74 (34.33)	35.21 (39.43)
CLEF+WP+EPq	32.41 (35.77)	35.29 (39.73)

Table 6: PSMT models with escaped named entities. Scores in brackets are BLEU scores without considering the actual NE translation.

As we can see, we can further improve the models using only our tiny amount of parallel training data and obtain scores comparable to the Google baselines. On the other hand, for the combined training data we can see a negative effect of this approach. However, as the scores in brackets show, improvement might be possible with a more sophisticated NE detection and translation procedure. These scores are measured on translating the "NE templates" only without replacing variables with corresponding names and, therefore, can be seen as upper bounds for this method.

3.4 Source Language Reordering

Yet another idea for improving our models is to apply source language reordering techniques before training and translating. This is especially important in our case when translating English questions where the predicate is often split into an auxiliary and the infinite main verb is moved to the end of the question. That this is a serious problem could be seen at the following translations obtained by the models from the previous section:

When did Armenia become independent ?
* Wanneer stierf Armenia onafhankelijk ?
(* *When died Armenia independent ?*)

This error appears of course not because "did" and "die" are so similar to each other but because there are apparently many questions about people's deaths in our training data. The preference for links at similar positions causes the word aligner to select "did" as the alignment of "stierf", which in the end causes the error described above.

When did Shapour Bakhtiar die ?
Wanneer stierf Shapour Bakhtiar ?
At what age did Fernando Rey die ?
Op welke leeftijd stierf Fernando Rey ?

The success of "pre-ordering" the source language has already been shown in earlier studies (Collins et al., 2005) and also moving the main verb in questions has been applied in other studies (Nießen and Ney, 2004). We therefore parsed our data with the Stanford parser obtaining not only phrase-structure trees but also dependency relations. We then moved the infinitive next to the auxiliary if they are in a (corresponding) direct relation to each other:

original: How did Jimi Hendrix die ?
reordered: How did die Jimi Hendrix ?

original: What language do the Berbers speak ?
reordered: What language do speak the Berbers ?

This is done for the CLEF questions before estimating the MT models and before translating questions from the test set. Results using this approach are shown in table 7.

	translation model	
language model	CLEF	CLEF+EP
CLEF	29.39	26.93
CLEF+EPq	33.18	38.07
CLEF+WP+EPq	33.58	37.46

Table 7: Simple re-ordering of the source language questions.

As we can see, the BLEU scores improve significantly for both, the small and the combined training data. Even for our small training set we now obtain scores above the Google baseline and for the combined data set we are more than five points ahead. This is very encouraging and further investigations in this direction should be carried out in future.

3.5 Analogical EBMT

Finally, we also want to look at the alternative approach of analogical learning for example-based MT. The approach has been briefly discussed earlier. We now apply it using the software of Lepage[4] and the CLEF questions together with our bilingual term database as example corpus. As expected the coverage of our examples is not sufficient. Only a small fraction could be translated using this technique (10 questions out of 100). It is not worth mentioning the BLEU score for the entire test set (the EBMT system functions as a translation memory returning the closest match in cases of failures of the analogical procedure). However, for the actual translations the accuracy in terms of BLEU scores is very high (70.7 BLEU). For a comparison, the best model so far from the previous sections scores only 66.1 BLEU on the same questions. It is therefore worthwhile considering this approach especially if the training corpus could be extended in future. We also experimented with a simple backoff approach in which we use the two-step procedure from section 3.3 together with the analogical EBMT approach in cases where the analogical solver did not succeed to find a translation of the original question. Using this strategy the number of translated questions goes up to 24. However, the BLEU score drops significantly to about 53.

4 Discussion & Conclusions

In this paper we addressed the task of translating questions for cross-lingual question answering. The motivation of this study is the development of a task-specific component that outperforms a general purpose engine. For this we used standard approaches to statistical MT with a mixture of task-specific data and out-of-domain data. One important aspect of our experiments is to test possibilities of building data-driven translation models from extremely scarce resources. Several techniques have been used ranging from source language reordering to named entity escaping and factored models with linguistic features. A summary of our results is shown in table 8.

According to the automatic measures on a small test set we succeeded to outperform a strong baseline given by a state-of-the-art general purpose translation engine (Google Translate). However, human evaluation should be performed in future to support the automatic evaluation. Furthermore, another look at the integration of linguistic features is also on our research agenda. Finally, we would

[4]We are very grateful for making this software available to us.

CLEF: 26.60, Europarl: 27.20, Google: 32.45

CLEF + extensions	
+ terms	28.53
+ terms & Q-LM	31.86
+ terms & Q-LM & escape NE	32.41
+ Q-LM & source-reordering	**33.58**
CLEF + EP + extensions	
Q-LM	**36.79**
Q-LM & factored	**36.18**
Q-LM & escape NE	**35.29**
Q-LM & source-reordering	**38.07**

Analogical EBMT (for 10 out of 100) = 70.67
backoff EBMT/NE (for 24 out of 100) = 53.06

Table 8: Summary of experiments. *Q-LM* refers to the language model trained on questions. Scores in bold denote results above the Google baseline.

also like to combine the strengths of the various approaches in order to build a system with better performance. Initial experiments with simple backoff strategies have already shown encouraging results.

References

Bouma, Gosse, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. 2005. Linguistic knowledge and question answering. *Traitement Automatique des Langues (TAL)*, 3.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistcal Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June.

Collins, Michael, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 531–540, Morristown, NJ, USA. Association for Computational Linguistics.

Curran, James R., Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the ACL 2007 Demonstrations Session (ACL-07 demo)*, pages 29–32, Prague, Czech Republic.

Frederico, M., N. Bertoldi, and M. Cettelo, 2008. *IRSTLM Language Modeling Toolkit, Version 5.10.00*. FBK-irst, Trento, Italy.

2008. Google translate. http://translate.google.com/.

Koehn, Philipp and Hieu Hoang. 2007. Factored translation models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic, June.

Koehn, Philipp, 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 ACL, demonstration session*, Prague, Czech Republic.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Larosa, S., J. Pearrubia, P. Rosso P., and M. Montes. 2005. Cross-language question answering: The key role of translation. In *Proc. Avances en la Ciencia de la Computacin, VI ENCuentro Int. de Computacin, ENC-2005*, pages 131–135, Puebla, Mexico.

Lepage, Yves and Etienne Denoual. 2005. Purest ever example-based machine translation: Detailed presentation and assessment. *Machine Translation*, 19(3-4):251–282.

Magnini, B., S. Romagnoli, A. Vallin, J. Herrera, A. Peas, V. Peinado, F. Verdejo, and M. de Rijke. 2003a. Creating the disequa corpus: a test set for multilingual question answering. In Peters, Carol, editor, *Working Notes for the CLEF 2003 Workshop*, Trondheim, Norway.

Magnini, B., S. Romagnoli, A. Vallin, J. Herrera, A. Peas, V. Peinado, F. Verdejo, and M. de Rijke. 2003b. The multiple language question answering track at CLEF 2003. In Peters, Carol, editor, *Working Notes for the CLEF 2003 Workshop*, Trondheim, Norway.

Magnini, B., A. Vallin, C. Ayache, G. Erbach A. Peas, M. de Rijke, P. Rocha, K. Simov, and R. Sutcliffe. 2005. Overview of the CLEF 2004 multilingual question answering track. In Peters, C., P. D. Clough, G. J. F. Jones, J. Gonzalo, M. Kluck, and B. Magnini, editors, *Multilingual Information Access for Text, Speech and Images: Results of the Fifth CLEF Evaluation Campaign*, volume 3491 of *Lecture Notes in Computer Science*. Springer.

Nießen, Sonja and Hermann Ney. 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2):181–204, June.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

van Noord, Gertjan. 2006. **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.

Developing Prototypes for Machine Translation between Two Sámi Languages

Francis M. Tyers
Departament de Llenguatges
i Sistemes Informàtics,
Universitat d'Alacant
E-03071 Alacant (Spain)
ftyers@dlsi.ua.es

Linda Wiechetek
Giellatekno,
Universitetet i Tromsø,
Norway
linda.wiechetek@uit.no

Trond Trosterud
Giellatekno,
Universitetet i Tromsø,
Norway
trond.trosterud@uit.no

Abstract

This paper describes the development of two prototype systems for machine translation between North Sámi and Lule Sámi. Experiments were conducted in rule-based machine translation (RBMT), using the Apertium platform, and statistical machine translation (SMT) using the Moses-decoder. The experiments show that both approaches have their advantages and disadvantages, and that they can both make use of pre-existing linguistic resources.

1 Introduction

In this paper we describe the development of two prototype machine translation systems between two Sámi languages, North Sámi (sme) and Lule Sámi (smj), one rule-based (Apertium), and one statistical (Moses). There are other systems which have been developed with marginalised languages in mind (e.g. (Lavie, 2008)), but, as of writing, these were not available under an open-source licence and thus could not be applied to the task at hand. The content will be split into several sections. The first section will give a general overview of the languages in question, and sketch a typology of MT scenarios for minority languages. The next sections will describe the two machine translation strategies in some detail and will outline how the existing language technology was able to be re-used and integrated. We will follow this by a short evaluation and then some discussion and future work.

1.1 The languages

Both North Sámi and Lule Sámi belong to the Finno-Ugric language family and are spoken in the north of Norway and Sweden, North Sámi also in Finland. North Sámi has between 15,000 and 25,000 speakers, while Lule Sámi has less than 2,000 speakers.

The Sámi proto-language was originally an agglutinative language, but North and Lule Sámi have developed features known from inflective languages (case/number combinations are often expressed by one suffix only, certain morphological distinctions are expressed by means of consonant gradation (i.e. a non-segmental process) only, etc.).

The main objective with the development of the prototype rule-based system was to evaluate how well existing resources could be re-used, and if the shallow-transfer approach was suited to languages with more agglutinative typologies.

1.2 A typology of MT systems for minority languages

Minority language speakers typically differ from the majority in being bilingual, the minority speaks the language of the majority, but not vice-versa. This has some implications for the requirements society will put to machine translation systems.

A majority to minority language system must be of high quality, so high that post-editing the output is faster than translating from scratch. The goal is to produce well-formed text, not to understand the content, since the minority language users will prefer the original to a bad translation. A minority to majority language system, on the other hand, will be useful even as a gist system, answering vital questions such as "what are they writing about me in the minority language newspaper?". The systems presented here are minority to minority language systems. North and Lule Sámi are mutually intelligible, and also in this context a gist sys-

Proceedings of the 13th Annual Conference of the EAMT, pages 120–127,
Barcelona, May 2009

tem will not be that interesting. The importance of the system lies in its ability to produce text. Here, North Sámi is the larger language, possessing close to a full curriculum of school textbooks. A high-quality MT system would help produce the same for Lule Sámi, and moreover from the closely related North Sámi than from Norwegian. The same situation may found for many language communities.

2 Rule-based machine translation

2.1 Apertium

Apertium is an open-source platform for creating rule-based machine translation systems. It was initially designed for closely-related languages, but has also been adapted to work better for less-related languages. The engine largely follows a shallow-transfer approach. Finite-state transducers (Garrido-Alenda and Forcada, 2002) and (Roche and Schabes, 1997) are used for lexical processing, first-order hidden Markov models (HMM) are used for part-of-speech tagging, and multi-stage finite-state based chunking for structural transfer (Forcada, 2006). The original shallow-transfer Apertium system consists of a de-formatter, a morphological analyser, the categorial disambiguator, the structural and lexical transfer module, the morphological generator, the post generator and the re-formatter.

2.1.1 Analysis and generation

For the analysis and generation, we used existing finite-state transducers for the two languages.[1]

lttoolbox has been widely used to model romance language morphology, and although it has been used to model the morphology of other languages with complex morphology (e.g. Basque), it is not ideal for these languages. *lttoolbox* is lacking features for dealing with stem-internal variation, diphthong simplification, and compounding.

North Sámi word forms involve both consonant gradation, diphthong simplification and compounding. The North Sámi noun *guolli* ('fish') alternates between *-ll-* (strong stage) and *-l-* (weak stage).

Additionally, one has to deal with diphthong simplification, the diphthong *uo* changes into a monophtong *u* in e.g. accusative plural *guliid*.

In the Apertium lexicon, *guolli* is represented as in figure 1. *gu* represents the stem, the item between `<l></l>` *liid* the generated ending and the items between `<r></r>` the analysis including the lemma and the morphological tags.

```
<pardef n="gu/olli__N">
 <e>
   <p>
     <l>liid</l>
     <r>olli<s n="N"/><s n="Pl"/>
                     <s n="Acc"/></r>
   </p>
 </e>
 ...
</pardef>
```

Figure 1: Section of inflectional paradigm for `gu/olli__N`

The Divvun and Giellatekno Sámi language technology projects[2] use finite-state transducers for the morphological analyser and closed-source finite-state tools from Xerox (Beesley and Karttunen, 2003). They tools handle two-level morphology model with twolc (two-level compiler) for morphophonological analysis together with lexical tools in a single transducer, consonant gradation, diphthong simplification and compounding are handled by two-level rules. Consonant gradation and diphthong simplification of the noun *guolli* are handled in the following way. *guolli* is listed in the root lexicon with lemma and continuation lexicon *AIGI* and redirected to the sublexicon *AIGI*.

```
guolli AIGI "fish N" ;

LEXICON AIGI !Bisyll. V-Nouns.
+N+Sg+Acc:%>X4 K ;
+N:%>X5 GODII- ; ! weak gr dipth simpl
...
```

From there it is redirected to a further sublexicon *GODII-* which redirects it to the sublexicon *GODII-*, which provides the plural accusative analysis.

```
LEXICON GODII-
 +Pl+Acc:jd9 K ;
```

At the same time, a two-level rule handles diphthong simplification when encountering the diacritical mark *X5* by removing the second vowel (*e o a*) in a diphthong (*ie uo ea*) if the suffix contains an *i*.

```
Vx:0 <=> Vow _ Cns:+ i (...) X5: ;
where Vx in (e o a) ;
```

[1] `http://giellatekno.uit.no/`

[2] To be found on `http://www.divvun.no/index.html` and `http://giellatekno.uit.no/`.

Consonant gradation is handled in another rule where a consonant (*f l m n r s ...*) is removed between a vowel, an identical consonant, another vowel, and a weak grade triggering diacritial mark (the rule is slightly simplified, noted by ...).

```
Cx:0 <=> Vow: _ Cy  Vow (...) WeG: ;
where Cx in (f l m n r s ...)
      Cy in (f l m n r s ...)
```

A general difficulty for generation and analysis are inconsistent tagsets in SL and TL. While verbs are specified with regard to transitivity (*V TV, V IV*) for North Sámi, they were not specified in the Lule Sámi dictionary (only *V*). Another matter of choice and convenience is the degree of lexicalisation as in the case of derived verbs. The North Sámi verbform *gohčoduvvo* ('he/she is called') either goes back to the form *gohčču t* ('order') or to *gohčodit* ('call, name'), which is derived from *gohčču t* but to some extent lexicalised.

```
gohčču t+V+TV+Der1+Der/d+V
       +Der2+Der/PassL+V+Ind+Prs+Sg3
gohčodit+V+TV+Der2+Der/PassL+V+Ind+Prs+Sg3

gåhtjudit+V+TV+Der1+Der/Pass+V+Ind+Prs+Sg3
```

In Lule Sámi, *gåhtjuduvvá* only gets the analysis with the lexicalised verb *gåhtjudit* as a lemma. The parallel derived form to North Sámi is not provided in the analysis. For the construction of the bilingual `sme-smj` dictionary, that means that *gohčoduvvo* is only matched with *gåhtjuduvvá* if *gohčoduvvo* is analysed with *gohčodit* as its lemma. In the bilingual dictionary, both pairs *gohčču t - gåhttjot* and *gohčodit - gåhtjudit* exist. But *gåhtjuduvvá* cannot be generated from *gåhttjot*.

```
<e><p><l>gohčču t<s n="V"/></l>
      <r>gåhttjot<s n="V"/></r></p></e>
<e><p><l>gohčodit<s n="V"/></l>
      <r>gåhtjudit<s n="V"/></r></p></e>
```

In the previous case, tag assymetry is due to annotation-choices. In other cases tag inconsistencies are linguistically motivated as in the case of the negation verb *ii/ij* ('not (do)'), which is specified with regard to tense in Lule Sámi, but not in North Sámi. This is due to the fact, that Lule Sámi has different present tense and past tense forms of the verb. North Sámi, on the other hand only has one form to express both present and past tense. The tense distinction is made by means of the main verb following the negation verb as in *ii boađe* ('he/she does not come') and *ii boahtán* ('he/she did not come').

```
ii        ii+V+IV+Neg+Ind+Sg3
ij        ij+V+Neg+Prs+Sg3
ittjij    ij+V+Neg+Prt+Sg3
```

Both for generation and analysis that means that one has to find a possibility to account for the 'missing' tag in North Sámi. 'Missing' means here the lack of tag specification for the *temps* (tempus) variable in the transfer files.

A number of multiword expressions differ from each other in SL and TL. While in North Sámi *gii beare* ('whoever') has inner inflection, the Lule Sámi *vajku guhti* does not. The initial pronoun *gii* corresponds to the second component *guhti* in Lule Sámi.

The last type of generation modification happens in a separate step. Orthographic variants and contractions are handled by the postgenerator. The Lule Sámi copula *liehket ('to be')* has three forms for the tag combination liehket+V+Ind+Prs+Sg3, *le, la, l*. While *le* and *la* are interchangable variants, *l* is a shortened form of *la* after wordforms that end in a vowel. The postgeneration lexicon specifies this change and outputs the correct form.

2.1.2 Disambiguation (Constraint Grammar)

Disambiguation of morphological and shallow syntactic tags is handled by the North Sámi parser. The parser uses Constraint Grammar, a formalism based on Karlsson (1990) and Karlsson (1995) and further developed by Tapanainen (1996) and Bick (2000).

The approach is bottom-up, which means that all input (ideally) receives one or more analyses. Those analyses are then one by one removed except for the last reading, which is never removed. The parser uses the output of the morphological transducer as an input and adds shallow syntactic tags. Syntax tags do not only function as the basis of a dependency tree structure representation, but also disambiguate morphology, e.g. homonymous genitive and accusative forms are distinguished on the level of syntax (genitive premodifier @→*N* vs. accusative object @←*OBJ*). The readings are then disambiguated by means of context rules.

The disambiguation file itself consists of different sections:

- **Sets**: lexical, POS, morphological features, syntactic, semantic lists one wants to abstract over
- **Syntactic annotation rules**: operators MAP and ADD annotate syntactic tags such as @←*OBJ*

- **Disambiguation rules**: operators SELECT and REMOVE either pick or discard a reading

In the Apertium engine, the Constraint Grammar module is added as a pre-disambiguator after the morphological analyser and before the statistical POS tagger. Apertium uses the r21668 version[3] of the parser, which is based on vislcg3.

A syntactic (or even semantic) analysis of the SL is also useful in MT, and the structural transfer in the `sme-smj` Apertium engine profits from syntactic information. By mapping the habitive tag *@HAB* onto locative nouns with habitive syntax/semantics, one can directly translate locative into inessive and a structural transfer rule in one of the MT modules becomes redundant. In the prototype system, the accuracy of the CG disambiguator has made the HMM-based tagger almost redundant.

It would appear that the rule-based Constraint Grammer parser is able to give a better performance than an HMM based tagger.[4]

Trigrams are not suitable for expressing syntactic structure.[5] CG on the other hand successfully expresses syntactic structure as a product of contextual disambiguation. (Bick, 2000, p.137)

2.1.3 Lexical transfer

Lexical transfer is handled in the bilingual dictionary, where entries have the form

```
<e><p><l>beaivi<s n="N"/></l>
      <r>biejvve<s n="N"/></r></p></e>
```

The North Sámi lemma with its POS specification comes first embedded in `<l></l>`, followed by the Lule Sámi lemma and its corresponding POS specification embedded in `<r></r>` In the case of a one-to-many relation between SL and TL, i.e. if several TL items exist for one SL item, the default translation is picked by means of the restriction `<e r="RL">`.

```
<e r="RL"><p><l>dàl<s n="Adv"/></l>
          <r>dàlla<s n="Adv"/></r></p></e>
```

[3] 2008-10-29 23:20:45 +0100

[4] Samuelsson and Voutilainen note in their comparison of a a linguistic and stochastic tagger that "at ambiguity levels common to both systems, the error rate of the statistical tagger was 8.6 to 28 times higher than that of EngCG-2." (Samuelsson and Voutilainen, 1997, p.251)

[5] According to Bick (2000), the syntactic structure problem is "unique" to probabilistic HMM grammars and resides in the "Markov assumption" that $p(tn|t1...tn-1) = p(tn|tn-1)$ (for bigrams), or $= p(tn|tn-1tn-2)$ (for trigrams).

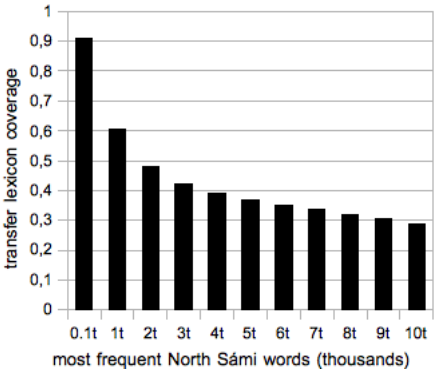


Figure 2: Coverage of the bilingual `sme-smj` dictionary

```
<e>       <p><l>dàl<s n="Adv"/></l>
          <r>dàl<s n="Adv"/></r></p></e>
```

A lexical selection module as described in the Apertium documentation (Forcada, 2008) is not employed by the system. Lexical transfer is considered to be regular instead of context-dependent. How close that is to the real situation is still to be decided.

The transfer lexicon was constructed in the following way: The orthographical differences between North and Lule Sámi are mostly regular. We thus made a finite state transducer which turned North Sámi lemmata into Lule Sámi candidates. The candidates were run through our Lule Sámi morphological transducer. Words recognised with the same POS as the input word were accepted, whereas words not recognised were manually revised. Semantic pairs which were non-cognates were manually added. Figure 4 shows the coverage of our transfer lexicon, for the n-thousand most common North Sámi lemmata.

2.1.4 Syntactic transfer

There are a number of structural differences between North and Lule Sámi that require structural transfer rules.

- The North Sámi locative case expressing place and source corresponds to either Lule Sámi inessive (place) and elative (source) depending on the context.
- In simple object constructions, the unmarked word order in Lule Sámi tends to be SOV, while it is SVO in North Sámi.
- In negation construction as discussed above, the Lule Sámi negation verb can inflect for

tense, while in North Sámi tense is expressed by means of the mainverb negation form

As the default translation of North Sámi locative (1) the Apertium system chooses Lule Sámi elative as in (2).

(1) son čokkii dávviriid ja dávttiid boares hávddi-in.
son čokkii dávviriid ja dávttiid boares hávddi-LOC.PL.
'(s)he collected things and bones old graves.from.'

(2) sån tjåkkij dávverijt ja dávtijt boares hávdi-js.
sån tjåkkij dávverijt ja dávtijt boares hávdi-ELA.PL.
'(s)he collected things and bones old graves.from.'

The default elative becomes inessive

- in habitive constructions,
- in place adverbials of stative verbs,
- before certain adverbs such as *gitta*.

In (3) a structural rule chooses inessive as a translation for locative when encountering the habitive tag *@HAB* distributed by a CG-rule, a verb from the verbs_stative list such as *ássat* ('live'), and an adverb from the ine_adv list such as *gitta* ('dependent on').

(3) Sámit dahjege sápmelaččat ásset Ruošša-s, Suoma-s ja Norgga-s.
Sámit dahjege sápmelaččat ásset Ruošša-LOC.SG, Suoma-LOC.SG ja Norgga-LOC.SG.
'Sámi or also 'sápmelaččat' live in Russia, Finland and Norway.'

(4) Sáme jali sábmelattja árru Ruossja-n, Suoma-n ja Vuona-n.
Sáme jali sábmelattja árru Ruossja-INE.SG, Suoma-INE.SG ja Vuona-INE.SG.
'Sámi or also 'sábmelattja' live in Russia, Finland and Norway.'

North and Lule Sámi differ with respect to word order. Especially in written texts, Lule Sámi allows for a number of unmarked SOV (6) construction whereas North Sámi prefers SVO (5).

(5) Anne ráhkada biepmu.
Anne makes food.

(6) Anne biebmov dahká.
Anne food makes.

Word order is treated in the second transfer module. The SOV rule in figure 3 captures the pattern (subject, verb, object) and outputs them in the order subject–object–verb by reordering the chunks indicated by *pos="1", pos="2"* and *pos="3"* into 1–3–2.

```
<rule>
  <pattern>
    <pattern-item n="SN_Subj"/>
    <pattern-item n="FMainV"/>
    <pattern-item n="SN_Obj"/>
  </pattern>
  <action>
    <out>
      <chunk>
        <clip pos="1" part="whole"/>
      </chunk>
      <b pos="1"/>
      <chunk>
        <clip pos="3" part="whole"/>
      </chunk>
      <b pos="2"/>
      <chunk>
        <clip pos="2" part="whole"/>
      </chunk>
    <out>
  </action>
</rule>
```

Figure 3: Transfer rule to convert SVO → SOV

The structural rules work successfully in transferring North Sámi to Lule Sámi structures. As the structural differences are minimal, the construction of rules is not very time-consuming. Rather the identification of structural differences is a new task as contrastive North-Lule Sámi grammar has been a rather neglected area within syntactic research.

3 Statistical machine translation

For the statistically based machine translation we used the Moses decoder, the word aligner GIZA++, and the srilm language model.[6]

3.1 Corpora

Minority languages may roughly be divided into three groups: The ones with a (limited) role in public administration or similar domains, the ones with a standardised written language and some text (more often than not the Bible comprises the bulk of the available corpus), and the ones with neither of these. Of our languages, North Sámi falls in the first group and Lule Sámi in the second. This means that the parallel resources available are extremely limited, they consist of the New Testament (approx. 150,000 words each), and a small corpus of school curriculum texts (appr. 15,000 words

[6]Available from the urls `http://www.statmt.org/moses/`, `http://www.fjoch.com/GIZA++.html`, `http://www.speech.sri.com/projects/srilm/` respectively

each, describing the content of the curriculum for the Sámi schools in Norway). The two NT versions have been translated in different countries (Norway/Finland and Sweden, respectively), with different Bible versions as source texts, and they differ from each other more than an ordinary parallel corpus would have done. The curriculum texts are probably translations of the same original – in any case the sentences are better matches of each other.

3.2 Training process

For the statistical machine translation, we build both factored and unfactored models. For Lule Sámi (the target language) we made both an unfactored and a factored trigram language model on our Lule Sámi corpus, 278,000 words. Half of the corpus (120,000 words) consists of New Testament (NT) texts, 106,000 belongs to the fact category, and 39,000 words is fiction. The factored model contained POS information, obtained from our Lule Sámi CG parser.

We then built various translation models. The models were severely limited by the availability of parallel corpora. We had one corpus consisting of the New Testament (9,200 parallel sentences), and one containing curriculum texts (1700 parallel sentences).

4 Evaluation

4.1 Qualitative evaluation

For the development of the Apertium system 16 test sentences from Wikipedia were used as regression tests.[7] Their target translations are based on a manual translation. Out of the 16 test sentences, 12 are successfully matched with the target translation at present. Remaining problems are not of a structural kind, but are dependent on one-to-many relations in the bilingual dictionary, tag inconsistencies between the `sme` and `smj` dictionaries, POS assymmetries and disambiguation errors from the Constraint Grammar disambiguator.

For evaluation purposes another independent manual translation is used. The Apertium translation deviates mostly with regard to lexical matters. Other lemmata were chosen. If they are synonyms or more idiomatic than the other ones remains to be studied. With regard to structural deviations, there was one deviating choice of case and one of

[7] `http://wiki.apertium.org/wiki/Northern_Sami_and_Lule_Sami/Regression_tests`

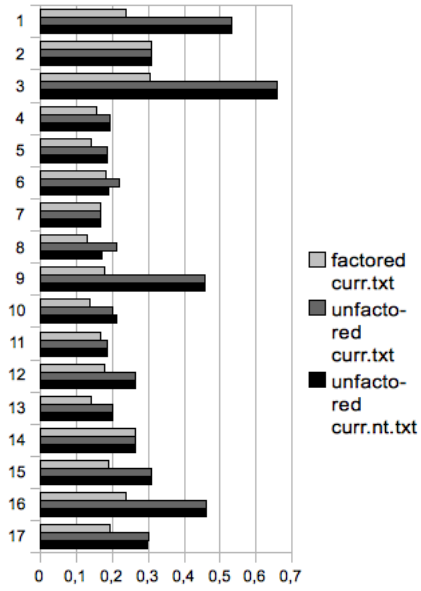


Figure 4: BLEU result for three models

word order. The word order deviation might hint at a rather optional SOV order.

The evaluation shows that while structural transfer seems to be mostly unproblematic, the choice of lexical tags and the lexical choices are the bigger challenge. Tags should be consistently chosen for both SL and TL whenever the deviation is not linguistically motivated. In the case of lexical choice, one needs to have a closer look at the bilingual lexicon. Are the deviations interchangable translations or is one of them the more idiomatic one?

4.2 Quantitive evaluation

For evaluation, we used the same 16 North Sámi Wikipedia test sentences (manually translated into Lule Sámi). For the SMT system, they were tested by three different translation models, a factored and an unfactored model based upon the curriculum corpus, and an unfactored model based upon the NT corpus (due to technical difficulties we were not able to make a factored model of the NT corpus).

The results are somewhat unexpected. Of the two versions of the curriculum translation model, the unfactored one is better than the factored one, with an average BLEU score of 0.3 as against 0.2. Comparing the two unfactored models, the larger one, containing NT and curriculum texts, performs similarly to the curriculum model for most sentences, but worse in some cases, resulting in a slightly worse overall score.

Type of deviation	Example
one-to-many relations	*dálla* vs. *dál* (both 'now')
tag inconsistencies	*iesjráddijiddje* ('self-governed') is analysed both as a deverbal form and a lexicalised adjective
POS assymetries	*gullujiddje* ('belonging') is analysed as a derived verb form
CG disambiguation error	*liehket* (infinitive) should be *li* (3rd person plural)

Table 1: Remaining transfer problems

Type of deviation	Example
lexical matters	*tjiehpe* vs. *smidá*, *moattegielak* vs. *ålogielak*, *sáhttá* vs. *máhttá*
case	*bargojn* vs. *bargoj*
word order	*manna l ulmmel* SVO vs. *man ulmmen la* SOV ('which is the purpose')

Table 2: Selection of divergences between North Sámi and Lule Sámi

Comparing the SMT and RBMT results is harder, as the lexicon for the rule-based system was small, and the grammar rule set was restricted. Thus, the RBMT did very well on known constructions (BLEU around 0.9 and better), but badly on new text. The SMT did badly across the board, and much of its success was due to the similarities of the languages (unknown words were passed through and now and then were correct).

With such a small training set, the result cannot be but bad. From earlier cross-linguistic research, a morphology-rich language such as Finnish comes out with clearly worse results than the more analytic German and French. Comparing BLEU score from (Banchs, 2005) with the token/-type ratio of Banchs' training set gives the picture in table 3.

	French	German	Finnish
Token/type	189	74	29
BLEU	0,302	0,245	0,203

Table 3: Token/type ratio and BLEU for 4 source languages in a Europarl MT study

The token/type ratio changes from genre to genre, but the relative distance between languages remain the same. This indicates that also an SMT system based upon a larger corpus would fare less than good for a morphologically complex language like Sámi.

5 Discussion

The corpora for Sámi are not good enough for SMT systems to be able to replicate the good RBMT results for North Sámi to Lule Sámi but much can be done both with tuning and corpus gathering. The corpora are probably good enough to build a gist system for North Sámi to Norwegian.

Apertium copes well with the structural transfer, but tag inconsistencies and many-to-many relations in the lexicon cause deviations between manual and automatic translations. A good lexicon and a consistent tagset are the basis for successful RBMT.

For morphologically complex languages, the lt-toolbox format for designing transducers might not be ideal, and one might consider other morphological transducers such as lexc and twolc.

Future plans in RBMT aim at making a full-coverage system out of the Apertium prototype. Word alignment can help constructing a more complete and better bilingual dictionary, and statistical methods could be used to choose the most idiomatic wordform in the case of one/many-to-many relations. Alternatively, a statistically-based lexical selection module as proposed in (Forcada, 2008) may be included. For optimisation of structural transfer, the Constraint Grammar could be enhanced by semantic roles that disambiguate between an inessive locative (PLACE) and an elative locative (SOURCE).

The available parallel corpora where Lule Sámi is one of the languages will not be large enough for SMT in the foreseable future.

Returning to the typology of MT systems for minority languages, we would like to explore the possibility of using SMT to create a gisting system for North Sámi to Norwegian. A corpus of 1,000 sen-

tences has already been tested. For this language pair, the linguistic distance is longer, but the empirical base far better (the present corpus collection contains appr. 120,000 sentences of parallel (but non-aligned) text). Although not much can be expected from a North Sámi–Lule Sámi SMT system, the development of a North Sámi–Norwegian system should be possible.

Acknowledgements

Many thanks to the anonymous reviewers for their helpful comments, and to Kevin Donnelly for reviewing an earlier version of this paper.

References

Banchs, Rafael E. and Crego, Josep M. and de Gispert, Adrià and Lambert, Patrik and Mariño, José B. 2005. Statistical Machine Translation of Europarl Data by using Bilingual N-grams, *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pp. 133–136

Beesley, K. R. and L. Karttunen. 2003. *Finite State Morphology* Vol. 1 CSLI Publications, Stanford. `http://www.fsmbook.com/`.

Bick, E. 2000. *The Parsing System 'Palavras': Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press, Aarhus.

Forcada, M. L. 2006. Open-source machine translation: an opportunity for minor languages. *Strategies for developing machine translation for minority languages*. 5th SALTMIL workshop on Minority Languages. pp. 1–7

Forcada, M. L. and B. Ivanov Bonev and S. Ortiz Rojas and J. A. Pérez Ortiz and G. Ramírez Sánchez and F. Sánchez Martínez and C. Armentano-Oller and M. A. Montava and F. M. Tyers. 2008. *Documentation of the Open-Source Shallow-Transfer Machine Translation Platform Apertium*. `http://xixona.dlsi.ua.es/~fran/apertium2-documentation.pdf`

Garrido-Alenda A. and M. L. Forcada. 2002. Comparing nondeterministic and quasideterministic finite-state transducers built from morphological dictionaries. *Procesamiento del Lenguaje Natural*. No. 29 pp. 73–80

Karlsson, F., Voutilainen, A., Heikkilä, J., and Anttila, A. (eds.). 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. Natural Language Processing* No. 4 *Mouton de Gruyter*, Berlin and New York.

Karlsson, F. 1990. Constraint Grammar As A Framework For Parsing Running Text. *Proceedings of COLING* Vol. 3 pp. 168–173

Lavie, A. 2008. Stat-XFER: A General Search-based Syntax-driven Framework for Machine Translation *Proceedings of CICLing 2008*, pp. 362–375

Roche, E. and Y. Schabes (eds.). 1997. Finite-State Language Processing. *MIT Press*, Cambridge, Massachusetts.

Samuelsson, C. and A. Voutilainen. 1997. Comparing a Linguistic and a Stochastic Tagger. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics* pp. 246–253

Tapanainen, P. 1996. *The Constraint Grammar Parser CG-2*. University of Helsinki Publications Vol. 27 pp. 246–253

Collocations in a Rule-Based MT System: A Case Study Evaluation of Their Translation Adequacy

Eric Wehrli, Violeta Seretan, Luka Nerima, Lorenza Russo
Language Technology Laboratory
University of Geneva
Switzerland
{firstname.lastname}@unige.ch

Abstract

Collocations constitute a subclass of multi-word expressions that are particularly problematic for machine translation, due 1) to their omnipresence in texts, and 2) to their morpho-syntactic properties, allowing virtually unlimited variation and leading to long-distance dependencies. Since existing MT systems incorporate mostly local information, these are arguably ill-suited for handling those collocations whose items are not found in close proximity. In this article, we describe an integrated environment in which collocations (and possibly their translation equivalents) are first identified from text corpora and stored in the lexical database of a translation system, then they are employed by this system, which is capable of dealing with syntactic transformations as it is based on a deep linguistic approach. We compare the performance of our system (in terms of collocation translation adequacy) with that of two major MT systems, one statistical, and the other rule-based. Our results confirm that syntactic variation affects translation quality and show that a deep syntactic approach is more robust in this sense, especially for languages with freer word order (e.g., German) and richer morphology (e.g., Italian) than English.

1 Introduction

Collocations, typical word combinations in a given syntactic relation (e.g., *warm greeting, distinct preference, [to] wreak havoc, [to] believe firmly, [to] break a record*) constitute a well-known problem for machine translation. Their identification in the source text and their proper processing by MT systems is the key factor in producing a more acceptable output (Orliac and Dillinger, 2003).

Over the past two decades, intensive efforts have been made to devise accurate techniques for collocation extraction from corpora; see, for instance, Church and Hanks (1990), Smadja (1993), Lin (1998), Evert (2004), among many others. Yet, the existing MT systems generally do not integrate collocational resources, or they are not designed to handle collocations in a specific and appropriate manner, as required by their high morpho-syntactic potential.

Therefore, such systems often achieve an unsatisfactory literal translation, especially when the collocation items are not found in the canonical order or in close proximity.[1] For instance, Example (1b) show the French translation returned by a major MT system, freely available online, for the English sentence in (1a), where the order of the verb and object of the collocation *break - record* is changed due to passivisation, and there are several words occurring between the two.

(1)a. *Records* are made to be *broken*.
b. *Les *dossiers* sont faites pour être *rompu*.
[The files are made for be broken.]

Since the system tested fails to identify that *records* and *broken* are part of a collocation, it is unable to propose a correct translation (in this case the French collocation *battre - record*), as it would normally do for a less problematic sentence, like *I want to break a record* (*Je veux battre un record*).

Another cause of failure appears to be the occurrence of collocations in atypical contexts: for

[1] According to Goldman et al. (2001, 62), as many as 30 words may intervene between the collocation items in a sentence.

Proceedings of the 13th Annual Conference of the EAMT, pages 128–135,
Barcelona, May 2009

instance, *give support* is correctly translated when found in a context like in Example (2a) (*give full support*), but not when it occurs in a less typical context like (2b) (*give massive support*).

(2)a. the people who rely on us to *give* full *support* when it is needed [...] → les gens qui comptent sur nous pour *apporter* leur plein *appui* quand il est nécessaire

b. and it is certainly right to *give* massive *support* to these areas [...] → et il est certainement droit de **donner* un *soutien* massif à ces domaines

Such examples indicate a high sensitivity of MT systems to the syntactic environment of the source collocations, which is clearly an issue given their marked syntactic flexibility. This further suggests that the collocation translation quality may seriously be affected for those source languages in which the word order is particularly free.

This paper describes the way collocations are treated in a large in-house machine translation system. The first condition for achieving an adequate translation for collocations is their accurate identification in the source sentence; in our system, this step is ensured by the detailed syntactic analysis provided by a deep syntactic parser.

Section 2 briefly states what exactly we mean by collocation, and indicates the challenges they pose to MT. Section 3 introduces our MT system, ITS-2, and provides details on its lexical database, as well as on the method used for extracting collocations (and their equivalents) from corpora. Section 4 describes the transfer method used by our system for translating collocations, then Section 5 presents an evaluation of the potential of our system to properly translate collocations.

2 Collocations

An agreed-upon definition of *collocations* does not exist yet; however, they are generally understood as a subtype of multi-word expressions that constitute arbitrary, conventional associations of words within a particular syntactic configuration.[2]

Unlike idioms, which exhibit either an opaque meaning—e.g., *to kick the bucket*, *to pull one's leg*—or very limited syntactic freedom, collocations have a fairly transparent meaning and are not subject to particular syntactic restrictions. Thus, a collocation of the type verb-object, such as *to break - record*, can be found in passive constructions, relatives or wh-interrogative clauses. Both of its components can undergo adverbial and adjectival modification, just like any verb and noun, as illustrated in Example (3):

(3)a. John *broke* the world *record*.

b. The world *record* has been *broken*.

c. The *record* that John *broke* was established in 2003.

d. In 1935, Jesse Owens set a long jump world *record* that was not *broken* until 1960 by Ralph Boston.

What makes collocations important for translation (and, in particular, for MT) is the fact that a large number of them do not translate well literally. It is therefore crucial to properly identify them and to dispose of the necessary bilingual resources to provide an adequate translation. The high frequency of collocations—several authors report a frequency of at least one collocation per sentence on average (Sinclair, 1991; Howarth and Nesi, 1996)—makes them a central issue in translation and motivates our particular interest in that matter.

In the remainder of the discussion we will restrict our attention to collocations of the verb-object type. This is one of the most common types of collocations, along with the adjective-noun type. At the same time, it is arguably the type that is the hardest to identify, due to the high frequency of extraposition of the object (as will be discussed in Section 4).

The non-identification of collocations dramatically affects the quality of the output. Collocations, which are in their vast majority semantically unambiguous (Yarowsky, 1993), are typically made of very common words, which in isolation may be polysemous (e.g., *break* in *break - record*). If the recognition of a collocation fails, the sense disambiguation information it carries is no longer available. This means that (even though a literal translation of collocations could in principle often result in an understandable if not fully adequate translation) the risk of choosing a wrong target word is rather high, making the literal translation option rather risky.

3 Our translation system

3.1 Overview

ITS-2 is a large-scale translation system developed in our laboratory, LATL, in the last couple of years (Wehrli, 1998; Wehrli et al., 2009). The language pairs currently supported are: English, German, Italian and Spanish to French, French-German, and French-English.

[2] See Heid (1994), Fontenelle (2001), Mel'čuk (2003), Grossmann and Tutin (2003) or Seretan (2008) for more detailed descriptions of the concept.

ITS-2 relies on an abstract linguistic level of representation, largely inspired from recent work in generative grammar (Chomsky, 1995; Bresnan, 2001; Culicover and Jackendoff, 2005). This level of representation is both rich enough to express the structural diversity of all the languages taken into account, and abstract enough to capture the generalizations hidden behind obvious surface diversity.

At the software level, an object-oriented design has been used, similar to the design adopted for the Fips multilingual parser on which it relies (Wehrli, 2007). To a large extent, ITS-2 can be viewed as an extension of the parser. It relies heavily on the detailed linguistic analysis provided by the parser for the supported languages, and exploits the lexical information of its monolingual lexicons. Both systems aim to set up a generic module which can be further refined to suit the specific needs of, respectively, a particular language or a particular language-pair.

The translation algorithm follows the traditional pattern of a transfer system. First, the input sentence is parsed by the parser, producing an information-rich phrase-structure representation with associated predicate-argument representations. The parser also identifies multi-word expressions such as idioms and collocations; this point is further detailed in Section 4.

Then, the transfer module maps the source-language abstract representation into the target-language representation. Given the abstract nature of this level of representation, the mapping operation is relatively simple and can be sketched as follows: recursively traverse the source-language phrase structure in the order: head, right subconstituents, left subconstituents. Lexical transfer (the mapping of a source-language lexical item with an equivalent target-language item) occurs at the head-transfer level (provided the head is not empty); it yields a target-language equivalent term, often (but by no means always) of the same category. Following the projection principle used in the parser, the target-language structure is projected on the basis of the lexical item which is its head.

However, the projections (i.e., constituents) which have been analyzed as arguments of a predicate undergo a slightly different transfer process, since their precise target-language properties may be in part determined by the subcategorization features of the target-language predicate. To take a simple example, the direct object of the French verb *regarder* in (4a) will be transferred to English as a prepositional phrase headed by the preposition *at*, as illustrated in (5a). This information comes from the lexical database. More specifically, the French-English bilingual lexicon specifies a correspondence between the French lexeme [$_{VP}$ regarder NP] and the English lexeme [$_{VP}$ look [$_{PP}$ at NP]]. For both sentences, we also illustrate the syntactic structures as built by the parser and/or the generator of ITS-2:

(4)a. Paul regardait la voiture.

b. [$_{TP}$ [$_{DP}$ Paul] regardait$_i$ [$_{VP}$ e$_i$ [$_{DP}$ la [$_{NP}$ voiture]]]]

(5)a. Paul was looking at the car.

b. [$_{TP}$ [$_{DP}$ Paul] was [$_{VP}$ looking [$_{PP}$ at [$_{DP}$ the [$_{NP}$ car]]]]]

3.2 The lexical database

The lexical database of ITS-2 is composed of several monolingual and bilingual lexicons. For each language supported by the underlying parser, the monolingual lexicons contain:

i) a table of lexemes, containing the base form and syntactic (as well as some semantic) information for words;

ii) a table of words, containing all the inflected forms for the entries in the table of lexemes;

iii) a table of collocations, which contains, in fact, multi-word expressions (including compound words and idioms as well).

For compound words, the storage structure used is the same as for simple words. Compounds are categorized, according to a lexical category, with their relevant syntactic features, and are recorded with all their inflected forms.

For collocational and idiomatic expressions, a uniform structure is used, which essentially contains the reference to the component words. Unlike compound words, collocations and idioms are assigned a syntactic category. The information stored in the lexicon of collocations includes:

- the type of syntactic relation that holds between the two components (lexical items[3]) of a collocation (e.g., noun-adjective, noun-noun, noun-preposition-noun, subject-verb, verb-object);

- the reference to the two lexical items composing the collocation;

- the preposition, when applicable;

- the frozenness features (plural collocation, determinerless complement, bare noun complement, etc.).

For instance, for the verb-object collocation *to take office* the lexicon entry contains the following information:

```
type:  verb-object
lexeme No.  1:  lex111038161 (take,
   transitive verb)
lexeme No.  2:  lex111026216 (office,
   common noun)
preposition:  ∅
frozenness features:
   bareNounComplement
```

As for the bilingual lexicons used by ITS-2, they contain source-target correspondences and information useful for the lexical transfer. For storage, a relational database management system was chosen. For each language pair, the bilingual lexicon is implemented as a relational table containing the associations between lexical items of the source language (SL) to lexical items of the target language (TL). The bilingual lexicon is bi-directional, i.e., it also associates lexical items of TL to lexical items of SL.

In addition to these links, the table contains transfer information such as translation context, preferences between one to many translations, semantic descriptors, and argument matching for predicates (mostly for verbs). The table structures are identical for all pairs of languages.

3.3 Collocation extraction

The number of collocations included in the monolingual and bilingual lexicons of our translation system varies from language to language, and it is currently on the order of several thousand entries. The French monolingual lexicon is the largest, with almost 13000 entries. We estimate that a number of 15000-20000 entries for each language would ensure an acceptable coverage. To achieve this coverage, we built a tool for collocation extraction from text corpora (Seretan, 2008), which we currently employ for discovering collocation candidates for inclusion in the lexicon.[4]

The tool provides advanced functionalities for visualizing the extracted results in their original context in the source corpora, and for managing a list of validated candidates to be added to the lexicon. The tool also integrates a sentence alignment module; therefore, whenever parallel corpora are available, the lexicographer can also visualize the target sentence and identify a translation equivalent for storing it in the bilingual lexicons.

The extraction of collocations from text corpora is done by using a hybrid extraction method, which combines syntactic information provided by our parser with existing statistical methods for detecting typical lexical associations in corpora. Thus, collocation candidates are first identified from each sentence based on the parse structures returned by the parser, as lexeme combinations in a given syntactic configuration (for instance, verb-object). Then, these candidates are ranked according to their probability to constitute collocations, as computed with the log-likelihood ratio association measure (Dunning, 1993). The tool also implements a wide range of other measures that the user can choose for ranking collocation candidates.

The method implemented is similar, in principle, to other hybrid methods that were lately applied for collocation extraction (Lin, 1998; Krenn and Evert, 2001; Orliac and Dillinger, 2003; Kilgarriff et al., 2004; Charest et al., 2007). By selecting candidate collocations as pair of lexemes in a given syntactic relation (such as head-modifier or predicate-argument), these methods are much more appropriate for handling flexible collocations than the standard syntactically-uninformed methods, which rely on the linear proximity of words.

Unlike the methods cited above, in our system the syntactic relations identified are "deeper" since the underlying parsing mechanism is more

[3]We call *lexical item* a lexeme or a collocation (more precisely, an entry in the table of lexemes or collocations). Note that through recursive embedding, a collocation may be formed of collocational subparts, as stipulated by theoretical studies, e.g., (Heid, 1994). For instance, *give full support* is made of two collocations, *give - support* and *full - support*.

[4]We believe that the insertion of new collocations in the lexicon cannot be done in a fully automatic way, as it is ultimately a lexicographer who must decide whether a group of words constitutes a collocation or not.

advanced, whereas the former make use of chunking, dependency parsing, or shallow parsing only. Thanks to the syntactic analysis performed by the parser, our extractor is also able to detect instances of a collocation even when they undergo complex grammatical operations, which are typical of constructions involving verbs. Also, with respect to the other extractors mentioned, our extractor has a broader grammatical coverage and supports a larger number of languages.

When parallel corpora are available in two languages which are both supported by the parser, a translation equivalent can automatically be detected for the extracted collocations with a method described elsewhere (Seretan, 2008). Experiments run on multiple corpora of several million words have permitted a substantial increase of collocation coverage in our lexical database.

4 Collocation translation with ITS-2

The translation of collocations in the ITS-2 system takes place in three phases: identification of a source language collocation, lexical transfer, and generation of a target language collocation.

Identification The proper identification of a collocation is arguably the most difficult task in our treatment of collocations. As we have shown, collocations of the verb-object type can occur in sentences in which the two lexemes constituting the collocation can be several words apart and not even in the expected order, due to syntactic processes such as passivization or *wh*-fronting. In extreme cases, the distance between the two lexemes can exceed several dozens of words (Goldman et al., 2001).

In order to adequately handle such sentences, a comprehensive syntactic analysis is necessary, capable of interpreting extraposed (fronted) elements and of resolving intra-sentential pronominal reference, as well as (at least some) extra-sentential pronominal reference. For instance, in order to identify the collocation *break - record* in Example (6a), the parser must be able (i) to recognize the presence of a relative clause, (ii) to determine the role of the relative pronoun with respect to the verb of the relative clause (direct object), and (iii) to identify the antecedent of the relative pronoun.

(6)a. The record that John has broken.

b. [$_{DP}$ the [$_{NP}$ record$_i$ [$_{CP}$ that$_i$ [$_{TP}$ [$_{DP}$ John] has [$_{VP}$ broken [$_{DP}$ e]$_i$]]]]]

This is what the parser does, returning a syntactic structure such as (6b), in which the index i shows the three-constituent chain connecting *record* with the direct object position of the verb *broken*.

One important function of a "deep" syntactic parser is to establish a syntactic normalization of the sentence, that is a canonical way of representing the fundamental structure of a sentence, abstracting away from the various surface structure differences due to grammatical (or stylistic) processes. Coindexed empty categories in argument positions or functional structures are examples of normalized structures commonly used in generative grammar.

With respect to the task of collocation identification, normalization is very helpful in the sense that it provides an abstract unified and standardized representation on which the presence (or the absence) of a collocation can be computed. To illustrate this point, consider the following example:

(7)a. The *deadline* that we had *set* could not be *met*.

b. [$_{TP}$ [$_{DP}$ the [$_{NP}$ deadline$_{i,j}$ [$_{CP}$ [$_{DP}$ e]$_i$ that [$_{TP}$ [$_{DP}$ we] had [$_{VP}$ set [$_{DP}$ e]$_i$]]]] could [$_{VP}$ not be [$_{VP}$ met [$_{DP}$ e]$_j$]]]]

As shown in structure (7b), the main subject *deadline* is the head of a double chain represented by the indices i and j, respectively. The first chain, i, expresses the relationship between the head of the relative clause and the direct object position of the embedded verb (as in the previous example), while the second chain, j, represents the fronting of the direct object of the main verb to the subject position, due to the process of passivization. Thanks to the normalization computed by the parser, the task of checking the presence of a verb-object collocation is therefore greatly simplified.

Transfer and generation Once a collocation has been identified in a source language sentence, all its members are marked as collocation members in order to prevent their automatic literal translation. Thus, the lexical transfer module will check in the bilingual lexicon whether an entry exists for that collocation. If not, the literal translation will apply. If yes, two different situations can arise:

1. The target language equivalent is a simple lexeme: in this case, the syntactic head of the collocation (in the case of a verb-object collocation, the verb) will be translated by means of that lexeme.

2. The target language equivalent is itself a collocation. This is what would happen in the case of the pairs *meet - deadline* and *set - deadline* in Example (7). For instance, for the first, our English-French bilingual lexicon specifies a correspondence between *meet - deadline* and *respecter - échéance*. Based on this information, *meet* will be translated as *respecter*, and the transfer module will take note that the lexical head of the argument corresponding to the direct object of the source language verb (in that case, also a direct object) will be the French lexeme *échéance*.

The transfer yields a target language abstract representation, to which grammatical transformations (e.g., passivization and other potential extraposition transformations) and morphological generation will apply to create the target sentence. Unless restrictions have been specified in the lexical database, collocations will undergo the exact same grammatical and morphological processes as other lexical items.

5 Evaluation

5.1 The experimental setting

A first evaluation experiment has been conducted to quantify the potential of the ITS-2 system to recognize collocations in the source text and to translate them correctly, and also to compare it against two state-of-the-art translation systems available online: *Google*, a statistical-based MT system,[5] and *Systran*, a rule-based MT system.[6]

The experiment consisted of manually evaluating the adequacy of the translations proposed by the three systems on a small test set of verb-object collocations. The manual evaluation was preferred over established MT evaluation metrics (such as BLEU) since we were interested here in a more focussed evaluation (i.e., the specific subtask of collocation translation evaluation), rather than in a global evaluation of sentence translation quality. Moreover, such metrics based on word-to-word matches are not really appropriate for collocation-oriented evaluation, as they underestimate the impact that the substitution of a single word (the collocate) has on the overall sentence quality.

Two source languages were considered, English and Italian, in order to allow cross-lingual comparison. The target language considered was French. The test set contains 200 collocation instances, half in English, half in Italian, that were attested in the English, and, respectively, Italian version of the Europarl corpus (Koehn, 2005).

The test set was built as follows. First, a number of 10 collocations of type verb-object has been selected in each source language, from among the results of our previous collocation extraction experiments. Their choice was motivated by the non-literal translation into French, the (supposed) high morpho-syntactic modification potential, and the sufficient occurrence in the corpus. The selected types are displayed in Table 1 (first column); the second column shows an adequate translation into French.

Collocation (English, Italian)	Translation (French)
bridge gap	combler lacune
draw distinction	établir distinction
foot bill	payer facture
give support	apporter soutien
hold presidency	assurer présidence
meet condition	remplir condition
pose threat	constituer menace
reach compromise	trouver compromis
shoulder responsibility	assumer responsabilité
strike balance	trouver équilibre
assumere atteggiamento	adopter attitude
attuare politica	mener politique
avanzare proposta	présenter proposition
avviare dialogo	entamer dialogue
compiere sforzo	consentir effort
dare contributo	apporter contribution
dedicare attenzione	accorder attention
operare scelta	faire choix
porgere benvenuto	souhaiter bienvenue
raggiungere intesa	conclure accord

Table 1: Collocation types in the test set.

Second, for each collocation type a number of 10 instances was identified in the Europarl corpus,[7] and the corresponding sentences were added to the test set. The method for choosing the instances was the following: the corpus documents were sorted in the reverse order of the document frequency of the noun (i.e., the object in each verb-object pair), then the first collocation occurrence was selected from each document.

The resulting test set was submitted to the 3 systems compared. Each of the 600 total sentences obtained was evaluated by two French native speakers, who performed a binary classifica-

[5] http://www.google.com/language_tools, accessed June 2008.
[6] http://www.systran.co.uk/, accessed June 2008.
[7] More precisely, only a subpart of the corpus was considered, namely the 2001 proceedings totalling 62 files and about 4 million words per language.

tion of the translation proposed for the source collocation:

1. correct - the translation corresponds to an adequate expression of the desired meaning in the target language;

2. incorrect - the opposite holds, i.e., either the meaning is not preserved, or it is preserved but the translation proposed is felt as unnatural/weird.

Table 2 shows the inter-rater agreement statistics for each subset <language pair, system>. The kappa statistic indicate a substantial inter-annotator agreement (0.69 on average). Despite this positive result, our analysis of disagreement cases indicated that the task of judging upon the acceptability of a collocation translation is not a trivial one, and that the context plays a very important role in the judgement.

	Language Pair	Google	Systran	ITS-2
Obs	English-French	87	86	88
	Italian-French	72	92	94
k	English-French	0.60	0.72	0.72
	Italian-French	0.42	0.82	0.85

Table 2: Inter-rater agreement: *Obs* – observed agreement, *k* – kappa statistic (Cohen, 1960).

5.2 Results and discussion

The precision of each system was computed as the ratio of correct translations to the number of consistently-annotated instances; the pairs on which the judges disagreed were discarded (their number is quite low, as can be seen from Table 2).

The precision achieved by each system for each language pair is displayed in the first two rows of Table 3. On the English data, our system is outperformed by Google (which is unsurprising, given that the Europarl corpus is extensively used by statistical MT systems for training),[8] but performs better than Systran (which is penalized by the insufficient coverage of its collocation lexicon). However, on the Italian data, our system outperforms both Google and Systran by a large margin. Whereas the performance of these systems dramatically degrades when switching the source language, that of our system remains stable (it is actually slightly better for Italian than for English).

[8]On the other hand, ITS-2 fails to identify some collocation instances and therefore to propose an appropriate translation. A preliminary error analysis has shown that this happens when the source sentences are particularly complex and the parser cannot build its complete syntactic analysis.

The next rows of Table 3 display the precision obtained when the test set is split in 3 disjoint subsets, according to the distance between the items of a collocation instance: low (distance=1,2); medium (distance=3,4) and high (distance>4).

	Language Pair	Google	Systran	ITS-2
all	English-French	83.9	52.3	71.6
	Italian-French	66.7	30.4	74.5
low	English-French	83.3	48.2	77.0
med	English-French	91.3	66.7	60.0
high	English-French	50.0	33.3	57.1
low	Italian-French	74.5	32.2	81.0
med	Italian-French	57.9	25.0	55.6
high	Italian-French	33.3	33.3	69.2

Table 3: Evaluation results: precision.

The values obtained show that the precision of all systems varies highly with distance, as well as from one source language to another. The collocation instances from the medium-distance subsets (i.e., those that allow 2 or 3 intervening words, like *meet - condition* in *meet the same conditions*, *conditions need to be met*) are those that are better handled by Google and Systran systems in English. In Italian, the systems appear to deal better with the low-distance subset (e.g., *sforzi compiuti*, *compiuto notevoli sforzi*). However, the three systems perform worse on the high-distance subsets. The decrease in precision is, nonetheless, lower for ITS-2: the maximal difference on subsets is 19.9%, whereas for Google is as high as 41.3%, and for Systran it is 33.4%.

This result indicates that the translation of collocations is indeed sensitive to the number of words intervening between the components items, and that beyond 3 words the precision deteriorates drastically. Our test set was, however, not balanced with respect to distance; rather, the distribution reflects the situation of a random sampling (due to the manner in which we built the test set). In the current configuration, only 9% of instances belong to the high-distance subset (while 25.5% belong to the medium-distance set, and 65.5% to the low-distance set). More investigation is needed on larger, balanced data in order to fully confirm the hypothesis that our deep syntactic approach is less affected by distance.[9]

[9]The choice of the test corpus, Europarl, might also have an influence on the reported results, as long as the Google system used the very same corpus for training. Future evaluation on a different corpus should provide more realistic results for this system; nonetheless, the results of the current evaluation will at least serve as upperbound reference for future experiments.

6 Conclusion

In this paper we showed how collocations are treated in ITS-2, a rule-based translation system. We argued that the quality of their translation depends in the first place of their successful identification in the input text, and this benefits, in turn, from the fine-grained syntactic analysis provided by a deep parser. At least as far as verb-object collocations are concerned, their identification is a true challenge for MT systems, since they can undergo a wide range of syntactic transformations.

A case-study comparative evaluation was performed on English-French and Italian-French data against two major MT systems available online. The results showed that i) all three systems perform worse when 3 or more words occur between the collocation items; ii) ITS-2 reaches the highest precision for the verb-object collocations for which the distance between the verb and the object is high (see Table 3, rows 5 and 8); iii) moreover, ITS-2 achieves the best precision for Italian, while the precision of the other systems decreases dramatically when switching from English to Italian (Table 3, rows 1 and 2). The average precision of ITS-2 on both languages is 73.0%, i.e., slightly less than one competing system (75.3%), but higher than the other (41.4%).

Our present evaluation was specifically focused on the quality of translations obtained for verb-object collocations. In future work, this evaluation should be extended to a larger dataset, to other language pairs, other corpora, and other collocation types, in order to gain better insights on how sensitive MT systems are to the syntactic flexibility of collocation. Another possible avenue for future research is the combination of syntactic and statistical techniques, expected to yield better results than either of the two approaches alone.

Acknowledgements

This work has been supported in part by a Swiss National Science Foundation grant (no 100012-113864). The authors wish to thank Alexis Kauffmann for participating in the annotation task.

References

Bresnan, Joan. 2001. *Lexical Functional Syntax*. Blackwell, Oxford.

Charest, Simon, Éric Brunelle, Jean Fontaine, and Bertrand Pelletier. 2007. Élaboration automatique d'un dictionnaire de cooccurrences grand public. In *Proc. of TALN 2007*, pages 283–292, Toulouse, France.

Chomsky, Noam. 1995. *The Minimalist Program*. MIT Press, Cambridge, Mass.

Church, Kenneth and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46.

Culicover, Peter and Ray Jackendoff. 2005. *Simpler Syntax*. Oxford University Press, Oxford.

Dunning, Ted. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Evert, Stefan. 2004. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, University of Stuttgart.

Fontenelle, Thierry. 2001. Collocation modelling: from lexical functions to frame semantics. In *Proc. of the ACL Workshop on Collocation*, pages 1–7, Toulouse, France.

Goldman, Jean-Philippe, Luka Nerima, and Eric Wehrli. 2001. Collocation extraction using a syntactic parser. In *Proc. of the ACL Workshop on Collocation*, pages 61–66, Toulouse, France.

Grossmann, Francis and Agnès Tutin, editors. 2003. *Les Collocations. Analyse et traitement*. Éditions De Werelt, Amsterdam.

Heid, Ulrich. 1994. On ways words work together – research topics in lexical combinatorics. In *Proc. of EURALEX '94*, pages 226–257, Amsterdam, The Netherlands.

Howarth, Peter and Hilary Nesi. 1996. The teaching of collocations in EAP. Technical report, University of Leeds.

Kilgarriff, Adam, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. The Sketch Engine. In *Proc. of EURALEX 2004*, pages 105–116, Lorient, France.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. of MT Summit X)*, pages 79–86, Phuket, Thailand.

Krenn, Brigitte and Stefan Evert. 2001. Can we do better than frequency? A case study on extracting PP-verb collocations. In *Proc. of the ACL Workshop on Collocation*, pages 39–46, Toulouse, France.

Lin, Dekang. 1998. Extracting collocations from text corpora. In *First Workshop on Computational Terminology*, pages 57–63, Montreal, Canada.

Mel'čuk, Igor. 2003. Collocations: définition, rôle et utilité. In Grossmann, Francis and Agnès Tutin, editors, *Les collocations: analyse et traitement*, pages 23–32. Editions "De Werelt", Amsterdam.

Orliac, Brigitte and Mike Dillinger. 2003. Collocation extraction for machine translation. In *Proc. of MT Summit IX*, pages 292–298, New Orleans, U.S.A.

Seretan, Violeta. 2008. *Collocation Extraction Based on Syntactic Parsing*. Ph.D. thesis, University of Geneva.

Sinclair, John. 1991. *Corpus, Concordance, Collocation*. Oxford University Press, Oxford.

Smadja, Frank. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.

Wehrli, Eric, Luka Nerima, and Yves Scherrer. 2009. Deep linguistic multilingual translation and bilingual dictionaries. In *Proc. of the Fourth Workshop on Statistical Machine Translation*, pages 90–94, Athens, Greece.

Wehrli, Eric. 1998. Translating Idioms. In *Proc. of ACL-COLING*, pages 1388–1392, Montreal, Canada.

Wehrli, Eric. 2007. Fips, a "deep" linguistic multilingual parser. In *Proc. of ACL 2007 Workshop on Deep Linguistic Processing*, pages 120–127, Prague, Czech Republic.

Yarowsky, David. 1993. One sense per collocation. In *Proc. of ARPA Human Language Technology Workshop*, pages 266–271, Princeton.

Automatic Translation of Norwegian Noun Compounds

Lars Bungum
Department of Informatics
University of Oslo
larsbun@ifi.uio.no

Stephan Oepen
Department of Informatics
University of Oslo
oe@ifi.uio.no

Abstract

This paper discusses the automated translation of Norwegian nominal compounds into English, combining (a) compound segmentation, (b) component translation, (c) bi-lingual translation templates, and (d) probabilistic ranking. In this approach, a Norwegian compound will typically give rise to a large number of possible translations, and the selection of the 'right' candidate is approaches as an interesting machine learning problem. Our work extends the seminal approach of Tanaka and Baldwin in several ways, including a clarification of some fine points of their earlier work, adaptation to a more adequate machine learning framework, application to a Germanic language with a small speech community and very limited existing resources, and systematic experimentation along several dimensions of variation.

1 Background: The Task

Compounding is a productive feature of the Norwegian language (just as in other Germanic language), and because Norwegian compounds are written in a single word (i.e. as one blank-separated entity) such constructions pose a challenge to automatic translation.[1] Consider the examples in (1), where we use a centered dot ('·') to typographically indicate component boundaries both in Norwegian compounds and literal English glosses:

(1) a. anlegg·*s*·vei
construction·road
'construction road'

b. dokument·stabel
document·pile
'pile of documents'

c. brud·*e*·spore
bride·spur
'fragrant orchid'

Both examples (1-a) and (1-b) can be translated adequately from the translations of their component parts: in (1-a) the formative *-s-* joins together the two components, where in (1-b) the Norwegian compound merely is the juxtaposition of two independent 'words'.[2] In terms of aligning components during translation, the Norwegian surface order is preserved in (1-a) (the English translation being a regular noun – noun compound), while (1-b) reverses the order of the component parts—in a different English construction, using the prepositional marker *of*.[3] We will refer to the correspondences between compound parts across languages as *translation templates* (see Section 3 below), where (1-a) and (1-b), for example, instantiate the templates $\langle N_1 N_2 \rangle \rightarrow \langle E_1 E_2 \rangle$ and $\langle N_1 N_2 \rangle \rightarrow \langle E_2 \text{ of } E_1 \rangle$, respectively.

Examples (1-a) and (1-b) are within the scope of our method, while (1-c) is not. The translation *fragrant orchid* is not accessible merely by translating the component parts of the Norwegian *brudespore*, and we call (1-c) *non-compositional* for our purposes. Furthermore, we limit our discussion to Norwegian *nominal* compounds with exactly *two* components, i.e. source language (SL) forms of the type $\langle N_1 N_2 \rangle$. We approach the task of translating such compounds as a processing pipeline of (a) compound analysis, (b) component translation, (c) template instantiation, and (d) ranking of translation candidates.

The number of candidate translations grows with the fertility of each component and the overall number of translation templates. We treat the selection of the best candidate as a ranking problem, employing a Maximum Entropy (MaxEnt) machine learning approach, and using a wide

[1] The Google translation services, for example, arguably present the best-performing open-domain Norwegian – English MT system to date. Nevertheless, the Google SMT system has no provisions for productively formed compounds.

[2] We use the term *word* in a purely technical sense here, i.e. for an independent unit of translation. In terms of the morphological structure of Norwegian compounds, the predominant analysis is as the combination of two (uninflected) *stems* (or lexemes), with inflection applying after compounding.

[3] For this example, it would seem appropriate to analyze *pile* as a relational noun, which would make the *of* PP a complement to the head noun. But for the purpose of the present discussion, nothing much will hinge on the specifics of the internal syntactic structure of English translations.

Proceedings of the 13th Annual Conference of the EAMT, pages 136–143,
Barcelona, May 2009

range of so-called *features*, encoding both mono-lingual and bi-lingual information for each translation candidate. Various MaxEnt ranking models are trained on a hand-crafted gold standard of 750 Norwegian compounds and preferred translations, and evaluated by means of cross-validation. Using this method, the best-performing model was able to select the exact gold standard translation for unseen test data in well above 50% of all cases.

In the following, we review closely related earlier work (Section 2), sketch the selection of experimental data, available resources, and specifics of our approach (Section 3), lay out the design of our experiments (Section 4), present a wealth of empirical results (Section 5), and finally conclude with a critical discussion of our findings (Section 6).

2 Earlier Work

In investigating the automatic translation of Norwegian nominal compounds, our starting point is the influential approach of Tanaka and Baldwin—henceforth T&B—who explore various ways of translating Japanese nominal compounds into English and vice versa (Tanaka and Baldwin, 2003a; Tanaka and Baldwin, 2003b; Baldwin and Tanaka, 2004). Abstractly, our steps (a) to (d) as sketched above are all taken from T&B, but there are important differences in the specifics of our approach, as well as extensions beyond the results of T&B. Besides, our focus on another language pair (with severely more limited resources available on the Norwegian source language side), most of the relevant differences pertain to the ranking step, arguably the key component in obtaining high-quality translations.

Tanaka and Baldwin (2003b) suggest to rank candidate translations based on target language (TL) distributional properties, essentially corpus frequencies. They develop an interpolated measure CTQ ('Corpus-based Translation Quality'; see Section 4 below), essentially ranking candidate translations according to the probabilities of component parts—relative to construction type, i.e. the English side of each translation template—and the probability of the candidate as a whole. CTQ is the reflection of linguistic arguments pointing to the importance both of the quantitative occurrence of a compound *itself* in a corpus, as well as to the propensity of its component parts to form phrases (of a specific construction type).

To avoid stipulating CTQ interpolation weights, Baldwin and Tanaka (2004) turn to a machine learning approach, proposing the creative (but mathematically dubious) use of a Support Vector Machine (SVM) classifier for the ranking task. The main shortcoming of their use of the SVM is the non-conditional nature of the probabilistic model, i.e. much like with CTQ; the task is construed as separating 'good' from 'bad' translations *independent* of the original SL compound.[4]

At the same time, Baldwin and Tanaka (2004) introduce additional sources of information, viz. bi-lingual properties extracted from machine-readable dictionaries. Intuitively, these additional machine learning features aim to provide a measure of the strength of the translation relation holding between component parts, and of course to actually capture those cases where SL compounds are fully listed in the dictionary. Our work extends Baldwin and Tanaka (2004) in several ways. First, we deploy a *conditional* MaxEnt ranker (rather than a contorted SVM classifier), leading to a formally more adequate and more scalable machine learning framework. We explore additional feature combinations of mono-lingual and bi-lingual sources of information, and provide a systematic investigation into the relevance of analysis 'depth' (contrasting a tagger vs. a syntactic parser) in preprocessing the training corpus. Finally, we provide empirical results on the learning curves—with increasing amounts of mono-lingual training data—of our various methods.

While T&B have been the foremost source of inspiration for our work, earlier approaches to the compound analysis and translation problem include Rackow et al. (1992), who explore the translation of German compounds into English. While their task is quite similar, this work has its emphasis on the segmentation and analysis of SL compounds, although it proposes using corpus data (counts) to distinguish between the various candidate translations. From the available information, the approach was not fully implemented or evaluated empirically. Grefenstette (1999), translating German and Spanish compounds, shows how WWW counts can be used to rank candidates, although his experiments are confined only to compounds for which a translation exists in a bi-lingual dictionary.

[4] Baldwin and Tanaka (2004) report that, in their SVM experiments, most of their training runs failed to converge, i.e. did not result in a functional classifier. This observation may well be owed to their creative use of the SVM framework.

3 Methodology and Preparational Steps

We pursued a data-driven approach both in the selection of training and test compounds and in the discovery of bi-lingual translation templates. A balanced set of 750 Norwegian $\langle N_1 N_2 \rangle$ compounds were extracted from running text, hand-inspected, and manually translated into English. Translation templates were then 'read off' the translations (the gold standard).

3.1 Source Language Compound Selection

Candidate Norwegian nominal compounds were selected from a large collection of running text, comprised of the Norwegian segments of the Oslo Multilingual Corpus[5], and of the smaller LOGON corpus (Oepen et al., 2004). The text corpus was analyzed using the Oslo-Bergen Tagger (OBT) (Hagen et al., 2000), which assigns a special SAMSET ('compound') tag to candidate compounds (i.e. input tokens not in the system lexicon, where a segmentation into known components is possible). Out of a total of 2,7 million words, 37,058 instances were labelled as compounds and nominals, of which 22,339 were unique types.[6]

To gauge frequency of use, Internet searches (using the Yahoo API) were performed for each of the unique compounds, and from the 4946 types that acquired more than 10 hits, we selected 750 at random. Much like in the original T&B experiments, these randomly chosen compounds were organized according to three frequency bands (according to Yahoo hits), henceforth: the low, middle and high bands. To identify compound-internal structure and confirm the $\langle N_1 N_2 \rangle$ construction type, we applied the procedure of Johannessen and Hauglin (1996), which is available as an optional component in the OBT. During this step, candidates that were segmented into more than two parts or other construction types were rejected and replaced with new random samples from the original set of 4946 words.

3.2 Gold Standard and Templates

Our final selection of 750 Norwegian $\langle N_1 N_2 \rangle$ compounds was presented to a bi-lingual informant, alongside the results of look-up in a Norwegian – English dictionary (Eek, 2001). The informant could either accept the translation, replace it or add to it, and provide translations for the compounds that were not listed in the available dictionary, which was the case for 95,6% of the compounds. Although alternatives in the translation were permitted, the informant was *not* instructed to provide an exhaustive list of possible translations. This was preferred to limiting the number of translations to one in all cases (as is the case in the earlier T&B experiments), as this would imply the undesirable assumption that any Norwegian compound, independent of context, has one and only one correct English translation. Of the 750 final SL compounds, 444 are compositional in our sense, i.e. the gold standard translation is available, in principle, to our method. The experiments reported in Section 5 focus on this compositional sub-set.

All translations were inspected and generalized into translation templates, essentially syntactic alignment instructions. The two templates seen earlier—$\langle N_1 N_2 \rangle \rightarrow \langle E_1 E_2 \rangle$ and $\langle N_1 N_2 \rangle \rightarrow \langle E_2 \text{ of } E_1 \rangle$—were the by far most frequent ones. We arrived at a total of 20 templates, including possessive constructions (e.g. *kvinne·avis* – woman·newspaper – 'woman's newspaper'), variation of the prepositional link (e.g. *jakt·lykke* – hunting·luck – 'luck in hunting'), morpho-syntactic variation of the non-head component, and even the reversed $\langle N_1 N_2 \rangle \rightarrow \langle E_2 E_1 \rangle$ (*gartner·mester* – gardener·master – 'master gardener'). This latter template which was attested only once in the gold standard, was excluded from our experiments as non-productive.

3.3 Target Language Statistics

A central element in the ranking of candidate translations is mono-lingual frequency information about the target language. To sample appropriate statistics, three large corpora of English text were used as the basis for the ranking task. The British National Corpus (BNC), comprising 80M words, the AQUAINT (AQ) corpus consisting of 375M words and finally the North American News Text Corpus (NAN) totalling 350M, words were all processed through the second version of the RASP parser (Briscoe et al., 2006), to make it possible to not only gather statistics of word (co-)occurrences but to also take into account the specific construc-

[5] See `http://www.hf.uio.no/ilos/OMC`.

[6] Note that these figures do not accurately reflect the frequency of compounding in Norwegian, as the OBT lexicon includes a relatively large number of high-frequency compounds, including many fully transparent and compositional ones. Due to the current OBT architecture, these instances are no longer identified with the SAMSET tag.

tion types. The parsed results were indexed according to the various templates, so that occurrence statistics for the compounds, their component parts, and the TL template structure could be easily extracted. In Section 4 below, we define various machine learning features on the basis of this data, and in Section 5, we investigate the effects of increasing amounts of available TL training data.

3.4 Task Definition and Evaluation

Our task is to automatically translate compounds according to the method outlined earlier. Seeing that the search space (the set of candidate translations) is fully determined by the bi-lingual dictionary and set of bi-lingual templates, the main factor of variation in our investigation is the ranking method applied to picking the 'best' candidate. In our experiments, we apply various rankers and evaluate against the gold standard translations. More precisely, we report the success rate as the percentage of Norwegian compounds for which the highest-ranked translation candidate is identical to the gold standard translation (or, in case of multiple references in the gold standard, is a member of that set). For the machine learning experiments, we apply ten-fold cross-validation, i.e. train the ranker on 90% of the gold standard and evaluate on the remaining 10%, repeating this procedure for all ten distinct splits, and averaging success rates over all runs. Thus, no model is tested on compounds that were part of its training data.

4 Experimental Setup

Recall that for the actual translation of a given compound, its component parts are looked up in the bi-lingual dictionary, and each component translated into its English counterparts. We will refer to the fertility of each component as n_1 and n_2, where for our example (1-a) above, say, $n_1 = 22$ and $n_2 = 5$, i.e. there are 22 available translations for the noun *anlegg* and 5 for *vei*, respectively.

4.1 Preparatory Steps

All component translations are 'slotted' into the translation templates, resulting in a set of translation candidates. The total number of candidates is the cross-product of n_1, n_2, and the number of distinct templates (20, in our experiments). This is indeed one of the richer examples, and in our experiments the maximum number of translation candidates did not exceed a couple of thousand possible outcomes. For each translation candidate, a set of quantitative corpus data is extracted from the pre-processed and indexed TL corpus. These data are then used to rank the candidates, in various ways, either by means of the CTQ of Baldwin and Tanaka (2004), or as the input to the MaxEnt ranker. While in the former (heuristic) case the corpus data can be directly used for ranking and testing on the gold standard (there is no separate training step), the MaxEnt approach requires separate training and test data sets, which we address by ten-fold cross-validation over the gold standard.

The splitting up of compounds (using the optional OBT component mentioned earlier) and component translation was carried out as a preparational step, where each SL compound and its component parts with TL translations were indexed in an intermediate data structure.

4.2 Candidate Generation with Templates

It was a requirement in the implementation that the Norwegian compounds could be split up into two parts, both of which were nouns. For the English translation, however, it is accepted that one of the components be translated as multiple English words, as in example (2). To accommodate this variation, all TL frequency counts discussed below can in principle range over any TL phrase, as observed in any of the candidate translations are any of the 'slots' defined by our set of translation templates.

(2) hytte·tilsyn
cottage·supervision agency
'cottage supervision agency'

4.3 Ranking Baseline: Reference

For the ranking task, as a simple baseline (i.e. a measure of how the more refined ranking methods performed), a reference ranking based on only the frequency (in the available TL corpora) of the translation candidate *in full* was introduced. Of two candidates, such as 'down bag' vs. 'bag of down', the most frequent phrase would be chosen.

4.4 Corpus-based Translation Quality

A much stronger baseline, borrowed from Baldwin and Tanaka (2004), was used—the interpolated CTQ metric[7]—which extracts the frequency

[7]Baldwin and Tanaka (2004) give a slightly revised formalization for CTQ, as compared to the earlier version of Tanaka and Baldwin (2003b). Furthermore, in the earlier publication there is room for uncertainty as to whether each term, esti-

Mono-Lingual Features
CTQ
freq(E_1, E_2, t)
freq$(E_1, _, t))$
freq$(_, E_2, t)$
freq(E_1, t)
freq(E_2, t)

Table 1: Corpus-based MaxEnt features, where E_1 and E_2 denote English phrases 'slotted' in as the first or second element of a compound template t. Most often, E_1 and E_2 are single words.

counts from the target language corpus.

$$\text{CTQ}(w_1^E, w_2^E, t) = \\ \alpha p(w_1^E, w_2^E, t) + \beta p(w_1^E, t) p(w_2^E, t) p(t) \quad (1)$$

Equation 1, firstly computes the probability of two English words, w_1 and w_2 occurring as an instance of the template t, multiplied by an interpolating weight, α, then adds the product of the probability of w_1 as the first element in a construction licensed by template t and the probability of w_2 being the second element, respectively. An example would be the count of *machine translation* occurring as two nouns in a sequence (the template) divided by the total count of all template instances, added to how often *machine* is the first word of such couples, and *translation* is the second, to capture what words more often let themselves be combined in such compounds.

4.5 MaxEnt Basics: Mono-Lingual Features

The Maximum Entropy (MaxEnt) framework has been applied successfully to NLP tasks before (Ratnaparkhi, 1996; Ratnaparkhi, 1998; Mikheev, 2000; Charniak and Johnson, 2005; Velldal, 2008) in areas like parsing, sentence boundary detection, and PoS tagging, but notably (re-)ranking, for which it is also used in this paper. The various statistics for each translation candidate (which will be discussed in further detail below), can be used as features in a conditional MaxEnt model (the family of MaxEnt models is also commonly referred to as log-linear or exponential models).[8]

mated by maximum likehood over the training corpus, should be conditioned on t or not: Tanaka and Baldwin (2003b) discuss the terms as 'conditional' probabilities, but equation 1 suggests a non-conditional formalization (in contrast to, for example, $p(w_1^E, w_2^E|t)$). We implemented both variants and found the non-conditional CTQ to perform substantially better, hence restrict ourselves to this variant in the following. Just like T&B, we use $\alpha = 0.9$ and $\beta = 0.1$.

[8]Like Velldal (2008) and much other current work, we make use of the open-source TADM framework, see `http://tadm.sourceforge.net` (Malouf, 2002).

Bi-Lingual Features
freq$(E_1, E2|N_1, N2)$
freq$(N_1, N2|E_1, E2)$
freq$(E_1, E2, \rightarrow)$
freq$(E_1, E2, \leftarrow)$
freq$(E_1|N_1)$
freq$(E_2|N_2)$
freq$(N_1|E_1)$
freq$(N_2|E_2)$

Table 2: Bi-lingual features, extracted from the dictionary. N_1 and N_2 denote the first and second element of the Norwegian compound and E_1 and E_2 designate the English translations of these components in the current translation template.

Given a source language compound n, our model estimates the probability of a candidate translation e_i as the normalized dot product of a vector $\vec{f}$ of so-called features—arbitrary properties determined by so-called feature functions—and a vector $\vec{\lambda}$ of corresponding weights:

$$p(e_i|n) = \frac{\exp \sum_j \lambda_j f_j(e_i, n)}{\sum_{k=1}^{n} \exp \sum_j \lambda_j f_j(e_k, n)} \quad (2)$$

The search for the highest-scoring candidate can then be formalized as $\arg\max_{e_i} p(e_i|n)$, i.e. finding the translation candidate e_i that maximizes the conditional probability, given n. The machine learning task, then, is to find the vector $\vec{\lambda}$ that maximizes the (conditional) likelihood of the training distribution—a problem for which off-the-shelf solutions are available.

To avoid the stipulation of linear interpolation weights in CTQ, we defined a MaxEnt model with a feature set consisting solely of (log-)frequencies extracted from the target language corpus. For all MaxEnt models that were built, an additional binary feature identifying the template, which would inform the model on which template was the most frequent, was used. The mono-lingual features that were used are shown in Table 1.

4.5.1 MaxEnt with Bi-Lingual Features

In addition to the two experiments testing the difference between humanly estimated interpolation weights and the results of using a machine learning engine, the MaxEnt learner was also tested on a full feature set, with features also encoding information about the individual translation(s) of the source input, and not just the monolingual target language features of the translation candidate. Our bi-lingual feature set, extracted from the one Norwegian – English dictio-

nary available, is summarized in Table 2. In this model, bi-lingual features are added 'on top' of the mono-lingual ones.

These dictionary-based features indicate how often an English component E_1 or phrase E_1E_2 is counted as a translation of its Norwegian source. Because there can be multiple senses of an entry in the dictionary, a translation can have frequencies above 1, meant to capture what is a more likely translation for a given source word. In addition, frequencies of the translation candidates attested in the dictionary, regardless of the source are captured, as well as using the dictionary in both directions. In Table 2 the symbol '$\rightarrow$' indicates use of the dictionary in 'forward' direction (Norwegian – English), and '$\leftarrow$' the reverse direction.

4.6 Variation in Analysis Depth

The RASP analyzer was used for the preprocessing of the English language text corpora. RASP results were then searched by means of regular expressions, corresponding to the TL side of our translation templates, in order to extract the frequency of the various types of translations. In performing these queries, there is a choice as to whether to use RASP annotations only at the part-of-speech (PoS) level, or whether to inspect full phrase chunks. Consider the simplified examples (3) and (4), showing attachment of a 'for' PP either inside of an NP, or as a VP modifier instead:

(3) (VP (VB buy)
(NP (NNS books)
(PP (IN for) (NP (NN children)))))

(4) (VP (VB buy) (NP (NNS books))
(PP (IN for) (NP (NN children))))

If the regular expression used for counting occurrences of the $\langle E_2$ for $E_1\rangle$ template only inspected the PoS tags associated to each word, both (3) and (4) would match, resulting in a false positive count. A regular expression query requiring all template elements to be embedded inside an NP, on the other hand, would count only the first one. Seeing that RASP annotations are fully automated, where the syntactic layer is bound to have a higher error rate than the PoS layer, however, it is not *a priori* known which of the two strategies would yield better approximations of the actual counts. Variation of analysis depth, in this sense, is a dimension of variation to all experiments summarized in Section 5 below.

4.7 Variation in Corpus Size

The experiments were conducted using the corpora BNC, AQ and NAN (as mentioned in Section 3), where additional training data was added incrementally, starting with only the BNC, then adding AQ, and finally also adding NAN. The amount of training data used is another, orthogonal dimension of variation to the experimental results reported below.

4.8 Parameter Tuning — Implementation

The TADM MaxEnt toolkit allows the tuning of certain hyper-parameters to the estimation process. Feature weights can be smoothed using a so-called Gaussian prior, and relative or absolute tolerance thresholds can be applied in determining learner convergence. A large space of different combinations for these hyper-parameters was explored experimentally, but learner performance was relatively stable within substantial intervals around the TADM default values; no specific combination lead to significantly improved performance, when compared to the default hyper-parameters. Thus, all results reported here assume standard TADM settings.

5 Results

An overview of experimental results can be found in Table 3, where REF denotes the simple frequency baseline, CTQ the original T&B metric, ME_1 our mono-lingual MaxEnt model, and ME_2 the full MaxEnt model, including dictionary features. The results show a notable increase in performance as we go from REF- and CTQ-based ranking to MaxEnt ranking, and a smaller, yet significant increase as the bi-lingual features are introduced. The increase between REF and CTQ shows how the weighted information about the 'association strength' between single component corpus data and the translation candidate itself boosts performance; and the difference between CTQ and ME_1 shows that it helps to combine these data through a principled machine learning approach. The fully superior performance of the MaxEnt model with all features, finally, suggests that adding more information (by way of features) to the model increases performance further.

In the following few paragraphs, we discuss these results further, along the various dimensions of variation that we have set up for these experiments.

		REF		CTQ		ME_1		ME_2	
Corpora	Band	*Tagger*	*Parser*	*Tagger*	*Parser*	*Tagger*	*Parser*	*Tagger*	*Parser*
BNC	high	28.03	25.00	32.58	31.82	39.80	38.70	51.10	51.90
	middle	20.51	19.23	26.28	33.33	31.80	36.30	51.20	50.90
	low	12.10	11.46	19.11	24.20	33.60	31.80	45.90	48.90
	all	19.77	18.20	25.62	29.65	34.81	35.42	49.3	50.49
+AQ	high	38.64	35.61	40.91	40.91	49.80	54.90	57.40	59.70
	middle	23.72	25.00	30.77	36.54	39.00	41.10	52.00	54.20
	low	13.38	12.10	19.11	20.38	26.80	27.80	45.50	46.70
	all	24.50	23.59	29.66	32.12	37.90	40.5	51.31	53.18
+NAN	high	35.61	37.12	38.64	38.64	49.40	51.60	58.70	59.60
	middle	23.08	24.52	26.92	29.03	38.60	39.60	51.80	52.20
	low	16.56	14.01	18.47	17.20	25.80	26.50	48.00	45.50
	all	24.50	24.55	27.42	27.70	37.28	38.54	52.51	52.03

Table 3: Overview of gold standard results, measured as the percentage of correctly translated compounds.

Frequency Bands In the success figures of Table 3, there is a general tendency across ranking methods to perform better on high-frequency compounds, presumably because frequency of use will impact the reliability of statistics used in ranking. We have not investigated this effect in a systematic manner, but recall from Section 3 that (a) the frequency bands were established from web counts (we lack a Norwegian corpus of sufficient size) and (b) our compound discovery procedure using the Oslo-Bergen Tagger is biased, in that a large number of compositional but frequent compounds have been entered into the system lexicon (as simplex words) and, hence, are omitted from our study. Thus, results presented here probably under-estimate the actual performance of our method.

Analysis Depth Table 4 shows the differences in performance between using tagger-based and parser-based data. For the three ranking methods displayed in the table, the parser-based generally data show an improvement in performance, i.e. the added precision of counts taking into account syntactic structure seems to outweigh the expectation of a higher error rate in RASP results at this higher depth of analysis. For all ranking methods, however, the difference is smallest when all training corpora are used, and parser-based counts even yield a slightly lower performance for the full corpus using all MaxEnt features (i.e. our most advanced model).

Corpus Size As Table 3 indicates, the performance of the various rankers generally increases as the base corpus from which quantitative data are extracted is larger. But it is also evident that going from the BNC to the BNC+AQ combination shows the biggest difference in performance. In fact, going from there to +NAN surprisingly indicates a decrease in performance, except for one set of experiments. The difference, however, is very small for the the most sophisticated ranking method, the fully-featured MaxEnt model. For 38 Norwegian compounds the top-ranked translation candidate diverged for the +AQ and +NAN experiments, with half of them going in either direction. Hence, a sign test exploring the likeliness of this result if the two methods +AQ and +NAN are equal, would find such an outcome expected, if the 'methods' are equal.

6 Discussion

Our experiments show that the MaxEnt approach is viable to finding the correct translation of nominal compounds, just as Baldwin and Tanaka (2004) show how a SVM can give better results than humanly stipulated interpolation weights. The performance also increases as a full feature set is used, including translation counts for the individual compound subparts, instead of only frequencies of the translation candidate itself.

The MaxEnt approach allows just for this combination of features, both features stemming from linguistic insight, as well as purely quantitative measures resulting from counts from annotated corpora. It will be possible to introduce further semantic information into such a model, when available, depending on the framework in which it is implemented. In our experiments, only one bilingual dictionary was used (Eek, 2001), but the

Corpora	REF	CTQ	ME_1	ME_2
BNC	-1,65	3,80	0,53	1,17
+AQ	-1,01	2,35	2,73	1,9
+NAN	0,14	0,28	1,3	-0,4

Table 4: Difference in performance when RASP is used as a parser and a tagger. A negative figure shows that tagger-based counts led to better ranking results.

counts for a translation could vary because of the different senses of one word stored in a lexicon entry. There may, however, also be other systematic relations between a compound and its correct translation, for example a relationship between a certain joint element and the output construction type, or the between semantic information and construction type. Such features could be implemented through the use of binary features, allowing them to be included in a MaxEnt model.

Although a larger corpus would likely yield better coverage of rare constructs, and accordingly help overall performance, a decrease in marginal benefit from adding words would also be expected. The low frequency band benefits less from the enlargement of the corpus, whereas the middle and high frequency bands show a marked improvement going from BNC to BNC+ANC. Our expectation was that the lower frequency band would benefit more from better coverage in the basis corpus, so this was an unexpected result. More research is needed to verify or explain this tendency.

References

Baldwin, Timothy and Takaaki Tanaka. 2004. Translation by Machine of Complex Nominals: Getting it right. In *Proceedings of the ACL04 Workshop on Multiword Expressions: Integrating Processing*, Barcelona, Spain.

Briscoe, Ted, John Carroll, and Rebecca Watson. 2006. The Second Release of the Rasp System. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australia.

Charniak, Eugene and Mark Johnson. 2005. Coarse-to-fine n-best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of the 43rd Annual Meeting of the ACL (ACL05)*, pages 173–180, Ann Arbor, MI, USA.

Eek, Øystein, editor. 2001. *Engelsk stor ordbok: engelsk – norsk/norsk – engelsk ('English Large Dictionary')*. Kunnskapsforlaget, Oslo, Norway.

Grefenstette, Gregory. 1999. The World Wide Web as a Resource for Example-Based Machine Translation Tasks. In *Translating and the Computer 21: Proceedings of the 21st International Conference on Translating and the Computer*, London, UK.

Hagen, Kristin, Janne Bondi Johannessen, and Anders Nøklestad. 2000. A Constraint-based Tagger for Norwegian. In *17th Scandinavian Conference of Linguistics*, Odense, Denmark.

Johannessen, Janne Bondi and Helge Hauglin. 1996. An automatic analysis of norwegian compounds. In *Papers from the 16th Scandinavian Conference of Linguistics.*, Turku, Finland.

Malouf, Rob. 2002. A Comparison of Algorithms for Maximum Entropy Parameter Estimation. In *Sixth Conf. on Natural Language Learning*, pages 49–55, Taipei, Taiwan.

Mikheev, Andrei. 2000. Tagging Sentence Boundaries. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 264–271, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Oepen, Stephan, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, and Victoria Rosén. 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, USA.

Rackow, Ulrike, Ido Dagan, and Ulrike Schwall. 1992. Automatic Translation of Noun Compounds. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 1249–1253, Nantes, France.

Ratnaparkhi, Adwait. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In Brill, Eric and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142. Association for Computational Linguistics, Somerset, New Jersey, USA.

Ratnaparkhi, Adwait. 1998. Maximum Entropy Models for Natural Language Ambiguity Resolution. Technical report, University of Pennsylvania.

Tanaka, Takaaki and Timothy Baldwin. 2003a. Noun-noun Compound Machine Translation: A Feasibility Study on Shallow Processing. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan.

Tanaka, Takaaki and Timothy Baldwin. 2003b. Translation Selection for Japanese-English Noun-Noun Compounds. In *In Proceedings of Machine Translation Summit IX*, New Orleans, LO, USA.

Velldal, Erik. 2008. *Empirical Realization Ranking*. University of Oslo, Oslo, Norway.

Marker-based Filtering of Bilingual Phrase Pairs for SMT

Felipe Sánchez-Martínez
Dept. Llenguatges i Sistemes Informàtics
Universitat d'Alacant
E-03071 Alacant, Spain
fsanchez@dlsi.ua.es

Andy Way
NCLT, School of Computing
Dublin City University
Dublin 9, Ireland
away@computing.dcu.ie

Abstract

State-of-the-art statistical machine translation systems make use of a large translation table obtained after scoring a set of bilingual phrase pairs automatically extracted from a parallel corpus. The number of bilingual phrase pairs extracted from a pair of aligned sentences grows exponentially as the length of the sentences increases; therefore, the number of entries in the phrase table used to carry out the translation may become unmanageable, especially when online, 'on demand' translation is required in real time. We describe the use of closed-class words to filter the set of bilingual phrase pairs extracted from the parallel corpus by taking into account the alignment information and the type of the words involved in the alignments. On four European language pairs, we show that our simple yet novel approach can filter the phrase table by up to a third yet still provide competitive results compared to the baseline. Furthermore, it provides a nice balance between the unfiltered approach and pruning using stop words, where the deterioration in translation quality is unacceptably high.

1 Introduction

The state-of-the-art statistical approach to machine translation (MT) is the phrase-based model. Phrase-based statistical MT (PB-SMT) systems (Zens et al., 2002; Koehn et al., 2003) are based on the log linear model combination of several feature functions (Och and Ney, 2002), one of which is the *phrase translation probability* estimated after extracting bilingual phrase pairs from the parallel corpus.[1]

Bilingual phrase pairs are automatically extracted after computing the word alignments (Brown et al., 1993; Och and Ney, 2003). The set $\mathrm{BP}(s_1^J, t_1^I, A)$ of bilingual phrase pairs extracted from the word-aligned sentence pair $s_1^I = (s_1, \ldots, s_i, \ldots, s_I)$ and $t_1^J = (t_1, \ldots, t_j, \ldots, t_J)$ is defined as in (1) (Zens et al., 2002):

(1)

$$\mathrm{BP}(s_1^I, t_1^J, A) = \{(s_i^{i+n}, t_j^{j+m}) : \forall (i', j') \in A : i \leq i' \leq i+n \Leftrightarrow j \leq j' \leq j+m\},$$

where $A = \{(i, j) : i \in [1, I] \wedge j \in [1, J]\}$ is a set of pairs with the alignment information between the words in the source sentence s_1^I and the words in the target sentence t_1^J.

According to equation (1), all words within a bilingual phrase pair are consecutive and not aligned with words from outside the bilingual phrase pair. It is worth noting that bilingual phrase pairs may contain words that are not aligned at all, even at the beginning or the end of the phrase.

In order to make the extraction of bilingual phrase pairs computationally tractable, it is normally the case that only those possible pairs within a certain *n*-gram length are considered because the number of possible bilingual phrase pairs grows exponentially with the length of the sentences. In such cases, the amount of phrase pairs extracted from the whole training corpus may render the resulting translation table unmanageable in terms

[1] In the context of SMT, a phrase may be any sequence of consecutive words, not necessarily syntactic constituents.

Proceedings of the 13th Annual Conference of the EAMT, pages 144–151,
Barcelona, May 2009

of memory usage, even for large-scale system deployment.

The building of such large-scale systems is the norm for research groups participating in MT evaluations such as NIST or WMT, and it is generally fine to do this where once-off translation is required, such as in the bulk localisation scenario for large multinational software companies, for example.

However, where online 'on demand' translation is required, it is completely impractical to deploy the exact same systems in these workflows. Such situations include multilingual call centre scenarios, or where users from different languages want to interact in real time (consider a 'multilingual Facebook' scenario, for instance). In addition, the problem of access to information is increasing all the time in a world where both delivery and interface devices are changing massively to enable pervasive, on-the-move access to digital content. One such example is the CMU Transtac "eyes-free and hands-free" two-way speech-to-speech translation system for translation in the field between English–Iraqi Arabic and English–Farsi (Bach et al., 2007).

For all these reasons, therefore, many researchers have begun to investigate ways in which intelligible translations can be produced in real time. In this regard, we devised a simple yet novel filtering approach based on the "Marker Hypothesis" (Green, 1979) (cf. section 3). Essentially, we use linguistic information in the form of closed-class word lists to filter the set of bilingual phrase pairs, by taking into account the alignment information and the type of the words involved in the alignments.

The inspiration for the set of experiments carried out in this paper was that successful Example-Based MT (EBMT) systems (Nagao, 1984; Carl and Way, 2003) have been built using the Marker Hypothesis to segment source–target aligned sentence pairs into linguistically motivated bilingual chunks (cf. (Way and Gough, 2003; Gough and Way, 2004)). These systems have proven to be particularly useful where good translation performance is required with much smaller translation tables than are traditionally used in PB-SMT. For example, Groves and Way (2005a) showed that for a range of systems built with different amounts of data, on average the translation table of a PB-SMT system was about five times the size of the equivalent EBMT system. In a related paper, on a training set of 203K English–French aligned sentence pairs, Groves and Way (2005b) showed that seeding a PB-SMT system built using Pharaoh (Koehn, 2004a) with 403,317 EBMT alignments, a BLEU score of 36.43 was obtained, compared to a score of 37.53 with 1,732,715 phrase pairs built using Giza++ (Och and Ney, 2003).

The remainder of this paper is organised as follows. Section 2 reviews other research work that has also focused on the filtering of the bilingual phrase pairs. Then, in section 3 we describe our approach. Section 4 describes the experiments conducted on four language pairs and the results achieved. The paper ends with our concluding remarks together with avenues for further research.

2 Related Work

Previous approaches to filter the phrase pairs used in PB-SMT can be divided into two classes:

- those methods that filter the phrase table according to the text to be translated;
- those more general approaches that filter the set of bilingual phrase pairs extracted from the training corpus, or the translation table directly, without knowing in advance which texts are to be translated.

Our approach falls into this latter category (cf. section 3).

In the first group of approaches we find the work by Koehn (2004a), which performs a rudimentary analysis of the sentences to translate in order to minimise the number of entries in the phrase table to be loaded into memory. A more sophisticated approach is performed in (Badr et al., 2007), where the authors consider the relationship between different translation models to obtain a much smaller set of phrases associated with each sentence to translate.

The approach in (Lü et al., 2007) may be considered somewhat in-between the two filtering classes because while translation table pruning is performed at training time, the authors know in advance what text is to be translated once training is completed. Lü et al. (2007) use well-known information retrieval methods to select sentences from the training corpus that better match the domain of the test corpus. Then these sentences are used to optimise the distribution of the whole training corpus.

In the second group of approaches, Eck et al. (2005) sort the training sentences according to the frequency of unseen n-grams so as to select a reduced number of sentences for training SMT systems to run on small devices. They also propose the use of information retrieval methods for that purpose. Ma et al. (2007) use alignment-guided chunking to filter the size of the translation table by 78.6%, with a 2.93-point reduction (about 15%) in BLEU score for German–English experiments.[2] Somewhat more impressively, Johnson et al. (2007) perform significance testing (Agresti, 1996) to select the sentences to be used in the training phase. They report a 90% reduction in the phrase table on various language pairs of the Europarl (Koehn, 2005) parallel corpus without any reduction in the translation quality achieved, as measured by BLEU.

3 Marker-based Filtering of the Bilingual Phrase Pairs

All words in a language can be classified into two different categories, namely closed and open classes. Closed-class words (henceforth, *closed* words) such as prepositions, pronouns or articles, may be thought as the *core* words of a language, i.e. as the words providing the structure for well-formed sentences, but without any special intrinsic meaning. In contrast, open-class words (*open words*, such as nouns or verbs) may be thought of as the words which express the meaning of a sentence. This difference between closed words and open words explains why no new words are usually added to the set of closed words, while the set of open words can easily grow as a language evolves.

Having a set of words that provides the structure for the remaining words to express their meaning is known as the Marker Hypothesis (Green, 1979), which states that the syntactic structure of a language is *marked* at the surface level by a closed set of *marker* (closed) words.

As stated in the introduction, this paper is not the first approach that has used the Marker Hypothesis in MT. While most work has centred on building EBMT systems which relate source and target phrase pairs comprised of words (e.g. (Juola, 1994; Way and Gough, 2003; Gough and Way, 2004; Groves and Way, 2005a)), systems have also been successfully constructed where phrases consist of word–morpheme mappings (e.g. (Stroppa et al., 2006) for English–Basque, and (Labaka et al., 2007) for Spanish–Basque). Marker-based chunking still plays a significant role in the MATREX (Stroppa and Way, 2006; Hassan et al., 2007; Tinsley et al., 2008; Du et al., 2009) system, where the sets of marker words needed for bilingual chunking are extracted automatically rather than by assembling these by hand as its predecessors did (e.g. (Way and Gough, 2003; Gough and Way, 2004)).

To give the reader some idea of how marker words are used in practice in such systems, we revisit an example from (Groves and Way, 2005a), namely (2) (from (Koehn, 2005), Figure 2):

(2) that is almost a personal record for me this autumn!
⟶c' est pratiquement un record personnel pour moi , cet automne!

.

Once 7 sets of closed-class words (determiners, quantifiers, conjunctions, prepositions, wh-adverbs, possessive and personal pronouns, cf. (6) below) have been built for English and French, the marker words in (2) can be tagged, as in (3):

(3) <DET> that is almost <DET> a personal record
<PREP> for <PRON> me <DET> this autumn!
⟶<DET> c' est pratiquement <DET> un record personnel <PREP> pour <PRON> moi , <DET> cet automne!

Then using marker tag information (label, and relative sentence position), and lexical similarity (via mutual information), the marker chunks in (4) are automatically generated from the marker-tagged strings in (3):

(4) a. <DET> that is almost : <DET> c' est pratiquement
b. <DET> a personal record : <DET> un record personnel
c. <PREP> for me this autumn : <PREP> pour moi cet automne

Should they be required, the set of generalised templates in (5) can be derived automatically from the bilingual phrase pairs in (4):

[2] While these results do not appear in the paper, they were included in the presentation accompanying that paper. Thanks to Yanjun Ma for this clarification.

(5) a. <DET> is almost : <DET> est pratiquement

b. <DET> personal record : <DET> record personnel

c. <PREP> me this autumn : <PREP> cet automne

All of these resources, together with a bilingual lexicon (in later work, induced via Giza++), are brought to bear in translating new input strings to good effect.

In this work, our approach to filter the set of bilingual phrase pairs used to train a PB-SMT system is based on the Marker Hypothesis and the following intuitive idea: as closed words provide the structure and open words provide the meaning, accurate bilingual phrase pairs should have an alignment between the open words of the languages involved in the translation, while closed words may remain unaligned, as the syntactic structure changes from one language to another. This same idea was used by Sánchez-Martínez and Forcada (2009) to filter the bilingual phrase pairs used in the inference of shallow-transfer rules for the Apertium MT platform (Armentano-Oller et al., 2006).

We have explored two different criteria to filter the set of bilingual phrase pairs. The first one ("open words alig") discards those phrase pairs presenting open words, in one or both languages, not aligned with at least one open word of the other language. The second criterion ("open words alig+borders") is more restrictive, as it discards all phrase pairs discarded by the first criterion and also those phrase pairs whose first or last words are not aligned with any word of the other language—in this case no matter whether the first and last words in each language are closed words or open words. We experimented with this second criterion because, as a result of the bilingual phrase pairs extraction algorithm, unaligned words may appear at the beginning or the end of a phrase, and we wanted to test whether this introduces any noise in the translation table. Note that, after extracting the set of bilingual phrase pairs from a word-aligned sentence pair, two or more bilingual phrase pairs may only differ in that some of them contain unaligned words at the beginning or the end of the phrases.

4 Experiments

We tested the presented approach on the following language pairs: Spanish–English (`es-en`), English–Spanish (`en-es`), French–English (`fr-en`), and English–French (`en-fr`). We used the data distributed for the WMT09[3] Workshop on MT both for training and testing. Unfortunately, the test sets contain only one reference translation, which causes the scores obtained to be somewhat lower than might otherwise have been expected.

All the experiments were performed using the Moses open-source decoder for PB-SMT (Koehn et al., 2007) and the SRILM language modelling toolkit (Stolcke, 2002). Training was carried out as follows:

1. Word alignments were obtained using Giza++ (Och and Ney, 2003) and symmetrized in the usual way (Koehn et al., 2003).

2. Bilingual phrase pairs were extracted from the word-aligned sentence pairs.

3. Extracted phrase pairs were filtered following the approach presented in this paper, and then scored.

4. Weights were optimised using minimum error rate training (MERT) in the usual manner (Och, 2003).

With the two filtering criteria explained in Section 3, we tested different lists of words:

closed words: A list of closed words in each language is provided to the filtering algorithm. These lists contain determiners, prepositions, pronouns, coordinate and subordinate conjunctions, relative and possessive pronouns, and punctuation marks. They consist of 193 Spanish words, 174 French words and 185 English words. Examples include those in (6):

[3] `http://www.statmt.org/wmt09/`

(6) *English*:

⟨DET⟩: {the, a, some ...}
⟨PREP⟩: {on, at, in ...}
⟨PRON⟩: {you, he, she ...}

French:

⟨DET⟩: {le, la, les ...}
⟨PREP⟩: {sur, dans, par ...}
⟨PRON⟩: {vous, il, me ...}

Spanish:

⟨DET⟩: {el, la, los ...}
⟨PREP⟩: {de, para ...}
⟨PRON⟩: {yo, tú, usted ...}

closed words+vaux: In addition to the closed words discussed above, all inflected forms of auxiliary verbs and modal verbs in each language were used. The verbs considered are: *deber*, *haber*, *poder*, *querer* and *ser* for Spanish; *avoir*, *devoir*, *être*, *falloir*, *pouvoir* and *vouloir* for French; and *be* and *have* for English. In sum, they consist of 1,572 Spanish words, 490 French words and 201 English words. The large number of words in Spanish is due to inflected forms with enclitic pronouns attached.

stop words: With the aim of avoiding the need to manually build a list of closed words in each language, we tested the use of stop words automatically obtained from the training corpus (cf. (Stroppa and Way, 2006), who extract the required sets of marker words automatically from online dictionaries). The underlying assumption here is that closed words, as the core words of a language, are very frequent and, therefore, will appear in every list of stop words.

Table 1 shows for the two filtering criteria ("open words alig" and "open words alig+borders", see Section 3) and for the different lists of words explained above, the percentage of bilingual phrase pairs discarded and the translation performance of the resulting translation model as evaluated with BLEU (Bilingual evaluation understudy (Papineni et al., 2002)) and TER (Translation edit rate (Snover et al., 2006)).

In all cases the baseline system, i.e. when no filtering of the bilingual phrase pairs is done, performs better than our approach. However, a significant reduction in the number of phrase pairs (around 25% for `es-en` and `en-es`, and around 33% for `fr-en` and `en-fr`) can be achieved at the cost of a small loss in the translation performance (around 0.012 in BLEU for the former two pairs, and about 0.017 for the latter). That said, having conducted significance testing using bootstrap resampling (Koehn, 2004b), these reductions in translation quality are significant. Of course, greater reductions in the number of phrase pairs can be achieved at an even higher cost in terms of translation quality.

With respect to which is the best filtering criterion, the results in Table 1 show that, as expected, a greater reduction—on average around 9-10%—in the number of bilingual phrase pairs is achieved through the "open words alig+borders" criterion. Nevertheless, this greater reduction is at the cost of a higher loss in translation quality, although only around 0.5 BLEU points on average across the board.

As for the list of words used by the filtering algorithm, it can be concluded that using a list of closed words gives better performance than the remaining lists of words. For example, for `es-en`, around a quarter of the phrase pairs are filtered with just over one BLEU point reduction in performance. For `en-fr`, the gap is even larger, with a one third reduction in the size of the translation table and competitive MT performance.

However, note that in some language pairs, adding auxiliary and modal verbs to the list of closed words provided slightly better results. For all language pairs bar `es-en`, although fewer phrase pairs were filtered, MT performance was better when auxiliaries and modals were included in the set of marker words, although the differences in MT performance were not always statistically significant.

Compared with the translation performance achieved when using stop words, it becomes clear that the use of closed words provides better results. For `es-en` and `en-es`, around 50% relatively more phrase pairs are filtered using stop words, but performance decreases by up to 3 BLEU points compared to when closed words are using as the filter. For `fr-en` and `en-fr`, we observe a drop in performance of around 1.5 BLEU points when

		open words alig			open words alig+borders		
Lang. pair	**List of words**	**filtered pairs**	**BLEU**	**TER**	**filtered pairs**	**BLEU**	**TER**
es-en	baseline		0.2355	0.6416		0.2355	0.6416
	closed words	24.73%	0.2232	0.6570	34.80%	0.2170	0.6673
	closed words+vaux	23.72%	0.2188	0.6644	34.69%	0.2157	0.6675
	200 stop words	36.42%	0.1952	0.6889	46.34%	0.1921	0.6885
	100 stop words	36.59%	0.1991	0.6818	45.96%	0.1942	0.6882
	50 stop-words	31.63%	0.2090	0.6705	41.15%	0.2037	0.6766
en-es	baseline		0.2208	0.6588		0.2208	0.6588
	closed words	24.72%	0.2090	0.6701	34.71%	0.2032	0.6823
	closed words+vaux	23.69%	0.2112	0.6713	34.59%	0.2039	0.6796
	200 stop words	36.38%	0.1845	0.6975	46.24%	0.1807	0.7077
	100 stop words	36.57%	0.1888	0.6935	45.86%	0.1838	0.7021
	50 stop words	31.64%	0.2014	0.6826	41.98%	0.1943	0.6897
fr-en	baseline		0.2331	0.6476		0.2331	0.6476
	closed words	33.04%	0.2128	0.6700	41.26%	0.2072	0.6747
	closed words+vaux	30.74%	0.2130	0.6693	40.16%	0.2076	0.6763
	200 stop words	36.20%	0.1947	0.6882	47.08%	0.1878	0.6932
	100 stop words	34.61%	0.2027	0.6795	44.89%	0.1968	0.6825
	50 stop words	35.14%	0.2082	0.6742	44.31%	0.2029	0.6790
en-fr	baseline		0.2105	0.6993		0.2105	0.6993
	closed words	33.08%	0.1965	0.7125	41.20%	0.1928	0.7208
	closed words+vaux	30.75%	0.1990	0.7114	40.07%	0.1957	0.7155
	200 stop words	36.17%	0.1807	0.7297	46.97%	0.1760	0.7345
	100 stop words	34.65%	0.1865	0.7239	44.81%	0.1798	0.7352
	50 stop words	35.18%	0.1903	0.7244	44.24%	0.1885	0.7241

Table 1: For each language pair, percentage of bilingual phrase pairs discarded and translation performance, as evaluated by BLEU and TER, for the two filtering criteria explained in Section 3, and for the different lists of words used in the experiments.

200 stop words are used, but with only around 10% relatively extra phrase pairs being filtered.

5 Conclusion and Further Work

We have presented a simple yet novel approach that may be used to filter the bilingual phrase pairs extracted from the parallel training corpus for deployment in PB-SMT in situations where a smaller system footprint is required. Our approach is based on the Marker Hypothesis and on the intuitive idea that the open-class words in a bilingual phrase pair should be aligned because they are responsible for the meaning, while it is less costly for closed-class words to remain unaligned.

The approach was widely tested on four European language pairs using different lists of closed words and two different filtering criteria. The results show that more than one quarter of the bilingual phrase pairs can be ruled out at the cost of a small (yet statistically significant) loss in translation quality. Despite this drop in performance, it is clear that more and more real examples are coming to the fore where a smaller translation table is absolutely necessary, such as the integration of PB-SMT systems in mobile devices, or to enable online on-demand translation between speakers having no common language.

As for future work, we plan to test whether the results may be improved if prepositions are not considered as closed-class words when they are part of a phrasal verb. In these cases the preposition changes the meaning of the verb and, therefore, does not play the role of a closed-class word in terms of the Marker Hypothesis.

We will also test our approach for the translation from English to a non-European language

such as Chinese, Japanese or Hindi. Chinese is the more difficult language, since it lacks some markers that would help to identify when a noun phrase is started. Japanese is easier, since in the creation of the language (from Chinese), some markers were introduced to facilitate reading. Hindi, as is the case with all Indian languages, has a one-to-one mapping to English word classes, and so we are confident that similar benefits may accrue as for the European languages tested in this paper.

Finally, we plan to deploy our system in a multilingual chat environment with a well-known multinational software company, as well as develop a 'multilingual Facebook'-type demonstration system. It will be interesting to see to what extent our distinction between open and closed words proves particularly instrumental under such conditions.

Acknowledgements

We thank Mikel L. Forcada for his advice and encouragement in the generation of this paper. We also thank Yanjun Ma, Tsuyoshi Okita, Sandipan Dandapat, Rejwanul Haque and Sudip Naskar for fruitful discussions on the application of the Marker Hypothesis to Asian languages. We would also like to thank the anonymous reviewers for useful comments which served to improve our work. Finally, we would like to thank our funding agencies: the first author thanks the support of the Spanish Ministry of Education and Science (project TIN2006-15071-C03-01); the second author acknowledges the support from Science Foundation Ireland (`http://www.sfi.ie`) through grants 05/IN/1732 and 06/RF/CMS064.

References

Agresti, A. 1996. *An introduction to categorical data analysis*. Wiley, New York.

Armentano-Oller, C., R. C. Carrasco, A. M. Corb-Bellot, M. L. Forcada, M. Ginestí-Rosell, S. Ortiz-Rojas, J. A. Pérez-Ortiz, G. Ramírez-Sánchez, F. Sánchez-Martínez, 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*, volume 3960 of *Lecture Notes in Computer Science*, pages 50–59. Springer-Verlag, Itatiaia, Brazil, May.

Bach, N., M. Eck, P. Charoenpornsawat, T. Köhler, S. Stüker, T. Nguyen, R. Hsiao, A. Waibel, S. Vogel, T. Schultz, and A. Black. 2007. The CMU Transtac 2007 eyes-free and hands-free two-way speech-to-speech translation system. In IWSLT 2007*, Proceedings of the 4th International Workshop on Spoken Language Translation*, pages 29–36, Trento, Italy.

Badr, G., E. Joanis, S. Larkin, and R. Kuhn, 2007. *Manageable Phrase-based Statistical Machine Translation Models*, chapter Computer Recognition Systems 2 (CORES 2007). Book Series: Advances in Soft Computing, pages 437–444. Springer, Berlin, Germany, October.

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Carl, M. and A. Way, editors. 2003. *Recent Advances in Example-Based Machine Translation*, volume 21 of *Text, Speech and Language Technology*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Du, J., Y. He, S. Penkale, and A. Way. 2009. Matrex: the DCU MT system for WMT 2009. In *Proceedings of the Fourth Workshop on Statistical Machine Translation, EACL 2009*, page in press, Athens, Greece.

Eck, M., S. Vogel, and A. Waibel. 2005. Low cost portability for statistical machine translation based on n-gram coverage. In *MT Summit X*, pages 227–234, Phuket, Thailand.

Gough, N. and A. Way. 2004. Robust large-scale EBMT with marker-based segmentation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 95–104, Baltimore, MD.

Green, T. 1979. The necessity of syntax markers. two experiments with artificial languages. *Journal of Verbal Learning and Behavior*, 18:481–496.

Groves, D. and A. Way. 2005a. Hybrid data-driven models of machine translation. *Machine Translation*, 19(3–4):301–323.

Groves, D. and A. Way. 2005b. 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*, pages 183–190, Ann Arbor, MI.

Hassan, H., Y. Ma, and A. Way. 2007. MATREX: the DCU machine translation system for IWSLT 2007. In IWSLT 2007*, Proceedings of the 4th International Workshop on Spoken Language Translation*, pages 69–75, Trento, Italy.

Johnson, H., M. Joel, G. Foster, and R. Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic.

Juola, P. 1994. A psycholinguistic approach to corpus-based machine translation. In *Proceedings of the Third International Conference on the Cognitive Science of Natural Language Processing (CSNLP-94), [no page numbers]*.

Koehn, P. 2004a. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the 6th biennial conference of the Association for Machine Translation in the Americas*, pages 115–124, Washington, DC.

Koehn, P. 2004b. Statistical significance tests for machine translation. In *2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 388–395, Barcelona, Spain.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. `http://www.statmt.org/europarl/`.

Koehn, P., F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2003*, pages 48–54, Edmonton, AL., Canada, May-June.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Labaka, G., N. Stroppa, A. Way, and K. Sarasola. 2007. Comparing rule-based and data-driven approaches to spanish-to-basque machine translation. In *MT Summit XI*, pages 297–30, Copenhagen, Denmark.

Lü, Y., J. Huang, and Q. Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 343–350, Prague, Czech Republic.

Ma, Y., N. Stroppa, and A. Way. 2007. Alignment-guided chunking. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 114–121, Skövde, Sweden.

Nagao, M. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In Elithorn, Alick and Ranan Banerji, editors, *Artificial and Human Intelligence*, pages 173–180, Amsterdam, The Netherlands. North-Holland.

Och, F. J. 2003. Minimum error rate training in statistical machine translation. In *41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July.

Och, F. J. and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *40th Annual Meeting of the Association for Computational Lingustics (ACL)*, pages 295–302, Philadelphia, PA.

Och, F. J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, K., S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *40th Annual meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA., July.

Sánchez-Martínez, F. and M. L. Forcada. 2009. Inferring shallow-transfer machine translation rules from small parallel corpora. *Journal of Artificial Intelligence Research*, 34. (In press).

Snover, M., B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, "Visions for the Future of Machine Translation"*, pages 223–231, Cambridge, MA., August.

Stolcke, A. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, Denver, CO., June.

Stroppa, N. and A. Way. 2006. MaTrEx: the DCU machine translation system for IWSLT 2006. In *Proceedings of IWSLT 2006 Workshop*, pages 31–36, Kyoto, Japan.

Stroppa, N., D. Groves, A. Way, and K. Sarasola. 2006. Example-based machine translation of the Basque language. In *Proceedings of the 7th biennial conference of the Association for Machine Translation in the Americas*, pages 232–241, Cambridge, MA.

Tinsley, J., Y. Ma, S. Ozdowska, and A. Way. 2008. Matrex: the DCU MT system for WMT 2008. In *Proceedings of the Third Workshop on Statistical Machine Translation, ACL 2008*, pages 171–174, Columbus, OH.

Way, A. and N. Gough. 2003. *wEBMT*: Developing and validating an EBMT system using the world wide web. *Computational Linguistics*, 29(3):421–457.

Zens, R., F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *KI 2002: Advances in Artificial Intelligence: Proceedings 25th Annual German Conference on AI*, volume 2479 of *Lecture Notes in Computer Science*, pages 18–32, Berlin, Germany. Springer-Verlag.

Tree-based Target Language Modeling

Vincent Vandeghinste
Centre for Computational Linguistics - KULeuven
Leuven, Belgium
vincent@ccl.kuleuven.be

Abstract

In this paper we describe an approach to target language modeling which is based on a large treebank. We assume a bag of bags as input for the target language generation component, leaving it up to this component to decide upon word and phrase order. An experiment with Dutch as target language shows that this approach to candidate translation reranking outperforms standard n-gram modeling, when measuring output quality with BLEU, NIST, and TER metrics.

1 Acknowledgements

The development of this system and research is made possible by the STEVIN-programme of the Dutch Language Union, Project Nr. STE-07007, which is sponsored by the Flemish and Dutch Governments, and by the SBO-programme of the Flemish IWT, Project Nr. 060051.

2 Introduction

In this paper we describe an approach to target language modeling using large treebanks. This introduction starts with a description of the MT system for which this target language modeling component is intented and continues with a short description of related research.

In section 3 we describe the details of the target language modeling component and in section 4 we describe an evaluation experiment for this component. Section 5 draws conclusions and sketches future work.

2.1 System description

We are developing a data-driven hybrid approach towards machine translation, reusing as much as possible already existing tools and resources to set up an MT architecture much like a classic rule-based transfer system. Instead of manually designing the rules, we intend to derive them from large parallel and monolingual (uncorrected) treebanks.

The system requires a source language parser and a parallel treebank, aligned from the sentence level up to the word level (Och and Ney, 2003), including sub-sentential alignment (Tiedemann, 2003; Tinsley et al., 2007, Macken and Daelemans, 2008). To get a parallel treebank we parse both the source and target language components of parallel corpora à la Europarl (Koehn, 2005). Each tree pair, sub-tree pair or word pair presents an example translation pair, and becomes a dictionary entry. This way we are removing the conceptual distinction between a dictionary and a parallel corpus, like Vandeghinste (2007).

In a similar fashion, but making abstraction of the concrete words, we derive a set of transfer rules from the available alignments. A translation model is built by counting the frequencies of occurrence of all these alignments.

The source language sentence is syntactically parsed, and the parse tree (and its sub-trees) is matched with the source language side parse trees of the dictionary/parallel treebank. The retrieved target fragments are then restructured according to the information in the transfer rules resulting in a target language bag of bags, which is structured like a parse tree, but without implying any surface order in the daughters of each node. When larger units are retrieved from the dictionary, their surface order is preserved, implying that some nodes in the bag of bags are not bags but trees, with ordered daughters.

It is up to the target language generation component to determine the lexical selection (which translation alternatives are preferred) and optimal surface ordering using the target language treebank. It is this component which we describe and

Proceedings of the 13th Annual Conference of the EAMT, pages 152–159,
Barcelona, May 2009

evaluate in the rest of this paper.

When the system has generated a translation, it is up to the human post-editor to accept the translation or to correct it. For this purpose a web-based post-editing interface is being designed, which allows adding, deleting, substituting, and moving words and phrases. The post-editor can choose amongst several translation alternatives for the sentence, or for certain parts of the sentence. When a sentence is accepted the post-editing information is fed back into the system's databases, updating the weights of both the translation model and the target language generation model.

2.2 Related Research

The hybrid MT system described in the previous section is similar to the *Data-Oriented Translation* (DOT) approach, which was first proposed by Poutsma (1998) and further researched by Hearne (2005). DOT uses Data-Oriented Parse Trees (Bod, 1992), whereas we use either rule-based parsers based on a set of linguistic rules and a stochastic disambiguation component or we use stochastic parsers trained on a manually parsed or corrected treebank. The DOT approach only uses small corpora and a limited domain, whereas we intend to use large corpora and a general domain (news).

The target language generation approach is somewhat similar to the *feature templates* used by the translation candidate reranking component of Velldal (2007), although there are some important differences: Velldal's feature templates can have a higher depth, whereas the patterns we extract can be seen as context-free rewrite rules, only capturing information about a mother and its immediate daughters. This can be attributed to the fact that the LOGON system (Lønning et al., 2004) for which Velldal built the component is a limited domain MT system (Tourist information) whereas we intend to build a large domain system (News), so we are using much larger corpora. Storing information at a similar level as Velldal is not feasible with such large treebanks.

Furthermore, our system borrows ideas for combining target language fragments from the METIS-II system (Carl et al., 2008; Vandeghinste, 2008).

Our system is being implemented from Dutch to English and French, and vice versa. In the rest of this paper, we assume Dutch as the target language.

3 The Target Language Generation Component

This section describes the approach we use for target language modeling. In section 3.1 we describe the input this component expects, section 3.2 describes the training procedure and the preprocessing steps applied on the training data, and section 3.3 describes how the target language generation component actually works.

The target language generation component is based on a large target language treebank. The input is assumed to be a source language independent bag of bags, as all elements in this bag are coming from the target language side of the dictionary, and the structure of the bag of bags is mapped onto the target language structure through the dictionary and the transfer rules.

3.1 Bag of Bags as input

We define a bag of bags as a *set of sets*, or in our case, as a parse tree representing the target language sentence, in which for each node,[1] the surface order of the daughters of that bag is undetermined, representing all permutations of the list of daughters. It is up to the target language generation component to resolve these bags and come up with the best solution.

In figure 1 you find an example of a bag of bags in xml-format representing the Dutch sentence "*Zie ook het kaartje hieronder.*" [Eng: Also see the map below.]. A regular parse tree for this sentence is presented in figure 2. Figure 1 represents besides this sentence numerous ($2! \times 4! \times 2! = 96$) other surface strings, each a permutation of the words in the sentence.

Note that in figure 1 we left out some features in the `<bag>` tags of the bag of bags for clarity and presentational purposes. The bag of bags is exactly the same as the xml output of the syntactic parse for the same sentence generated by the Alpino parser (van Noord, 2006), apart from the fact that the `<node>` tags in the parse tree have been replaced by `<bag>` tags in the bag of bags, indicating that these bags still need to be resolved, and from the fact that it does not contain position information.

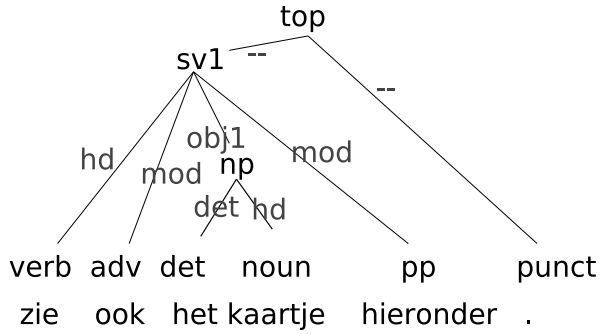
The Alpino parser is the parser we use for Dutch syntactic analysis. It is a parser which is based on head-driven phrase structure grammar (Pollard

[1] Some of the sub-trees are coming straight from the dictionary, so they are not sub-bags and do not need to be resolved.

Figure 1: An example bag of bags

```
<bag cat="top" rel="top">
 <bag cat="sv1" rel="--">
  <bag frame="verb(hebben,sg1,
              transitive_ndev_ndev)"
       pos="verb" rel="hd" word="Zie"/>
  <bag frame="sentence_adverb" pos="adv"
       rel="mod" word="ook"/>
  <bag cat="np" rel="obj1">
   <bag frame="determiner(het,nwh,nmod,
               pro,nparg,wkpro)"
        pos="det" rel="det" word="het"/>
   <bag frame="noun(het,count,sg)"
        pos="noun" rel="hd"
        word="kaartje"/>
  </bag>
  <bag frame="er_adverb(onder)" pos="pp"
       rel="mod" word="hieronder"/>
 </bag>
 <bag frame="punct(punt)" pos="punct"
      rel="--" word="."/>
</bag>
```

Figure 2: Parse tree for the example sentence (without frames)



and Sag, 1994) giving both phrase structure and dependency information.

Resolving the bag of bags in a bottom-up fashion, we first resolve the noun phrase (NP) "*het kaartje*" [Eng: the map]. There are two possible permutations for this NP, and we want to find the most probable. How this is done is explained in section 3.3.

When the NP is resolved, we need to resolve the *sv1*, which stands for *a sentence with the verb in first position*. The sv1 has four daughters, so this amounts to 24 (4!) different possible surface orders.[2] One of these daughters has two possible outcomes, so this already totals 48 translation alternatives under investigation.

This procedure is applied on all non-terminal bags.

[2]Because we treat all categories the same, we do not make use of the fact that for an *sv1* we know that the verb should be, by definition, in first position.

3.2 Training the target language generation component

In order to resolve the bags, we train the target language generation component on a large treebank. For Dutch, this treebank was automatically annotated by the Alpino parser (van Noord, 2006), and is available online at http://www.let.rug.nl/∼vannoord/trees/.

It consists, amongst others, of the following corpora: the Spoken Dutch corpus (CGN) (Oostdijk et al., 2002), the Lassy corpus (van Noord et al., 2006), the Dutch part of Europarl (Koehn, 2005), and the Dutch wikipedia.

The total corpus used in the experiments in section 4 consists of 290,658,861 words in 18,048,702 sentences, averaging 16.10 words per sentence.

From each of these sentences, we collect the *rewrite rules* at different levels of abstraction. For instance, for the example sentence "*Zie ook het kaartje hieronder*", we would collect the information in table 1.[3]

Note that we abbreviated some of the frames to fit in the table and that we use "|" as a field separator between the different kinds of information represented in our rewrite rules. Consecutive elements on the right-hand side of the rules are written with a space inbetween or on a new line. For instance, the `sv1` rule has four right-hand side symbols on every abstraction level.

We distinguish several different levels of abstraction, going from very abstract (Level 1: Relations) to very concrete (Level 7: Head + Frame/Cat + Relations).

1. Relations (Rel): Containing the dependency relations and the function information.

2. Part-of-speech/Category (Pos/Cat): containing the parts-of-speech of terminal nodes and the category for non-terminal nodes.

3. Pos/Cat + Rel: containing the combinations of parts-of-speech/category information and dependency information.

4. Frame/Cat: Containing frame information for terminal nodes and the category information for non-terminals. Frames are generated by the Alpino parser, and are a very fine-grained part-of-speech tag.

[3]This sentence has a parse tree exactly like the example bag of bags, apart from replacing the `<bag>` tags with `<node>` tags.

Table 1: Extracting information from a sentence at different abstraction levels

Level 1: **Relations**
top : -- -- sv1 : hd mod obj1 mod np : det hd
Level 2: **Pos/Cat**
top : sv1 punct sv1 : verb adv np pp np : det noun
Level 3: **Pos/Category + Relations**
top : sv1|-- punct|-- sv1: verb|hd adv|mod np|obj1 pp|mod np : det|det noun|hd
Level 4: **Frame/Category**
top : sv1 punct(punt) sv1 : verb(hebben,sg1,transitive...) sentence_adverb np pp np : determiner(het,nwh,nmod,pro...) noun(het,count,sg)
Level 5: **Frame/Category + Relations**
top : sv1|-- punct(punt)| sv1 : verb(hebben,sg1,transitive...)|hd sentence_adverb|mod np|obj1 pp|mod np : determiner(het,nwh,nmod,pro...)|det noun(het,count,sg)|hd
Level 6: **Head + Pos/Cat + Relations**
top : sv1|--|Zie punct|--|. sv1 : verb|hd|Zie adv|mod|ook np|obj1|kaartje pp|mod|hieronder np : det|det|het noun|hd|kaartje
Level 7: **Head + Frame/Cat + Relations**
top : sv1|--|Zie punct(punt)|--|. sv1 : verb(hebben,sg1,...)|hd|Zie sentence_adverb|mod|ook np|obj1|kaartje pp|mod|hieronder np : determiner(het,nwh...)|det|het noun(het,count,sg)|hd|kaartje

Table 2: Number of different labels and bags

Abstraction Level	Labels	Bags
1 Rel	32	50,233
2 Pos/Cat	48	568,299
3 Pos + Rel	510	1,584,535
4 Frame	36,729	9,764,647
5 Frame + Rel	50,130	10,251,079
6 Head + Pos + Rel	22,924,782	60,753,604
7 Head + Frame + Rel	26,400,004	61,283,814

5. Frame/Cat + Rel: containing the combinations of frame/category and relation information.

6. Head + Pos/Cat + Rel: containing the combination of the head word of a node with the parts-of-speech /category and relation.

7. Head + Frame/Cat + Rel: containing the combination of the head word of a node with the frame and relation.

In table 2 we present some information about our database for the total corpus size of 18 million sentences. The second column (Labels) indicates the number of different labels (types) for that abstraction level. The third column (Bags) shows the number of different bags at that level. If the corpus contains two or more permutations of the same bag, then these are counted as one bag.

All this data is collected over the whole treebank, and put in a database, precalculating which patterns are permutations of each other, and adding the frequency of occurrence for each of these permutations.

We have one database table per category per abstraction level, and we have 25 categories for Dutch, resulting in 175 tables. Each of these tables contains one row per bag and one column per subcorpus. For each bag and each corpus, we store the surface order of the bag elements and their frequency, allowing multiple surface orders and frequencies per database cell.

The use of separate columns for sub-corpora allows us to easily activate and deactive certain parts of the total corpus. It is a design choice that facilitates adapting the MT system to specific domains by activating the appropriate columns.

3.3 Matching the Bag of bags with the training data

We want to resolve the noun phrase-bag "*het kaartje*", knowing that there are two possible permutations.

We start of on the most concrete level, looking for the occurrence in the training data of either

```
np :  det(...)|det|het
      noun(...)|hd|kaartje
```

or

```
np :  noun(...)|hd|kaartje
      det(...)|det|het
```

If one or both of these occur in the training data, then we use their relative frequencies as weights for the solution. When neither of them occurs in the training data, we go to a more abstract level, hoping to find information regarding the relative higher occurrence of one permutation over the other, cascading over the different abstraction levels, until the bag is resolved. In the rare case that none of the abstraction levels can resolve the bag, all permutations get the same weight.

We use a set of cut-off parameters to limit the number of alternative analyses under consideration to a manageable number. Currently, we keep only track of the 10 best scoring alternatives. When no information or equal frequencies are found, and the bag would generate more than 30 permutations, we cut off at 30. This is especially required in the experimental conditions where the corpus size is still low (cf section 4). We stop processing an alternative solution if its weight is 10 times lower than the weight of the current best solution, and for each node, we allow a maximum of 100 combinations of the solutions of the daughters. As the system is currently fast enough, we have not yet investigated different values for these cut-off parameters, but it is clear that cutting off sooner would lead to faster processing but lower accuracy. Most of these cut-off parameters come in action only at low corpus sizes and/or in experimental conditions with only high abstraction levels.

4 Experiment

In this section we describe an experiment in which we evaluate the target language generation component of our MT system in isolation, excluding factors that might contribute to the translation quality in good or bad sense that are not part of the target language model.

Section 4.1 describes the methodology that is used for the experiment, and section 4.2 describes the evaluation results.

4.1 Methodology

In a way, we are translating from Dutch to Dutch, only evaluating the ordering mechanism used in the target language generation component.

We tested the quality of the output of the target language generation component by comparing it to the input sentence from which the bag of bags originates, which serves as a reference translation when evaluating with BLEU (Papineni et al., 2002), NIST (Doddington, 2002), and TER (Snover et al., 2006).

Additionally we also measured the number of exact matches: those cases in which the output sentence is identical to the input sentence.

We have constructed a test set of 575 real-life sentences from a real translation context that were parsed with Alpino and converted into bags.

We have several test conditions in two dimensions:

1. *Corpus size*: expressed in number of sentences. The treebank consists of several subcorpora, and we tested the system while gradually adding these sub-corpora. The size of these sub-corpora serves as data points on the X-axis in figures 3, 4, 5, 6, and 7.

2. *Abstraction level*: we have described the seven abstraction levels for Dutch in section 3.2. We tested the system with only the data for the most abstract level available, gradually adding less abstract levels. These are the data series 1 to 7 in the legend.

As a baseline, we also calculated the quality of a *trigram language models*. We used the SRILM toolkit (Stolcke, 2002) to train a backoff trigram model. Additional baseline testing with a fourgram model with Chen and Goodman's (1998) modified Kneser-Ney discounting did not yield better results. As it is not feasible to generate all permutations and then calculate their likelihood, we implemented a branch and bound approach. For each sub-bag, all permutations were generated and these were ordered according to their likelihood, keeping only the 10 best for each sub-bag. When any of these permutations contained more than n words, a sliding window of size n was used to estimate their likelihood. This procedure was recursively applied until the whole bag is resolved.

Figure 3: Effect of corpus size and abstraction level on BLEU score

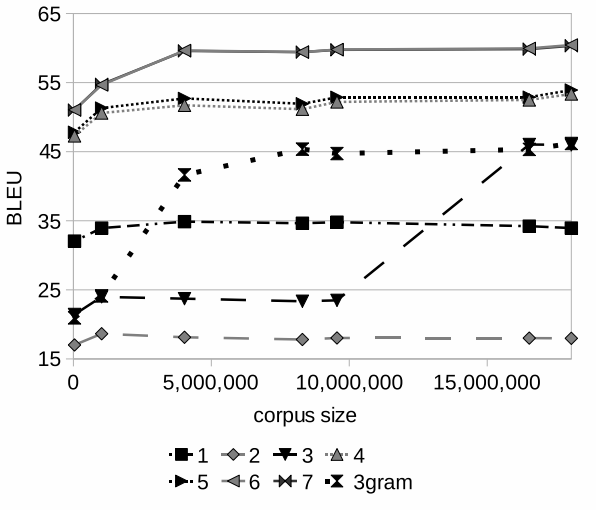


Figure 4: Effect of corpus size and abstraction level on NIST score

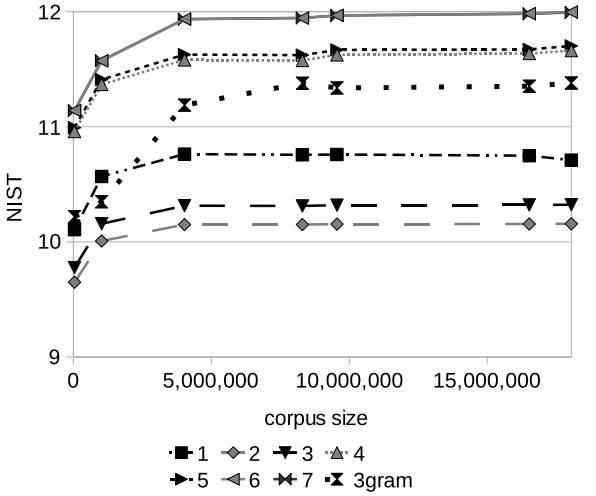


Figure 5: Effect of corpus size and abstraction level on TER score

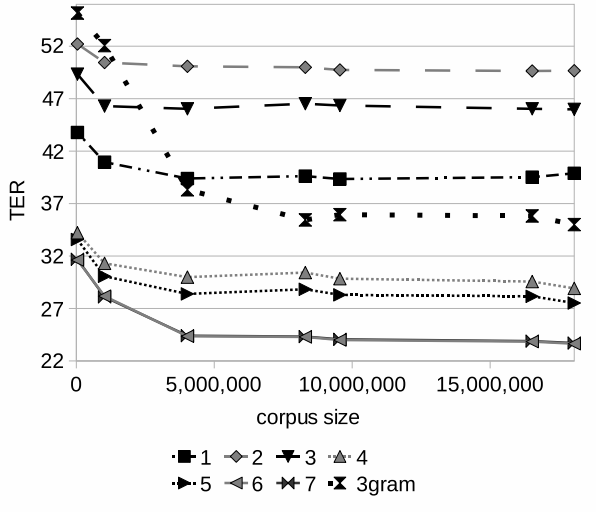
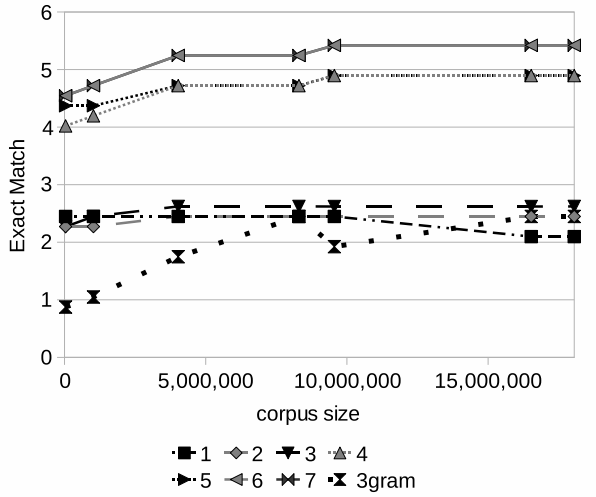


Figure 6: Effect of corpus size and abstraction level on percentage of Exact Matches



For exact match we calculated a baseline by counting the total number of possible permutations and the probability of ranodmly picking an exact match.

4.2 Results

When looking at figure 3 it is clear that the addition of the least abstract levels yields the best results, although there is not much difference between levels 6 and 7. At the largest corpus size, level 6 even outperforms level 7. This can be explained by the fact that there is only a relatively small difference in granularity between levels 6 and 7, which is clear when looking at table 2. There is a reduction of numbers of bags of less than 1%, so the abstraction is very limited. In future versions of the system, we might omit level 7 as it does not add any accuracy.

It is also clear that for all corpus sizes, abstraction levels 4, 5, 6, and 7 outperform the baseline.

The results are consistent for the NIST scores shown in figure 4.

When looking at the TER scores in figure 5, the same observations are still true. Note that TER expresses an error rate, so lower scores are better.

A somewhat unexpected result is the fact that level 1 consistently outperforms levels 2 and 3. We assume some kind of artefact and will investigate this further.

The percentage of exact matches, as presented

in figure 6 confirms the results from the other metrics. Note that the probability of randomly picking one of the possible permutations of the input bag of bag as its solution would result in an exact match baseline of 0.0000911%, so all experimental conditions improve over this baseline.

5 Conclusions and Future Work

We have set up a translation generation component for a parse and corpus-based MT system. This component requires a bag of bags as input, each bag and sub-bag representing all permutations of their respective daughters.

We trained the component on a large target language treebank (with fully automatic parses) so we can look up for each of the bags whether it occurs in the corpus, in what surface order, and with what frequency.

Comparing our system to a standard *n-gram* model we can conclude that our system clearly outperforms this baseline.

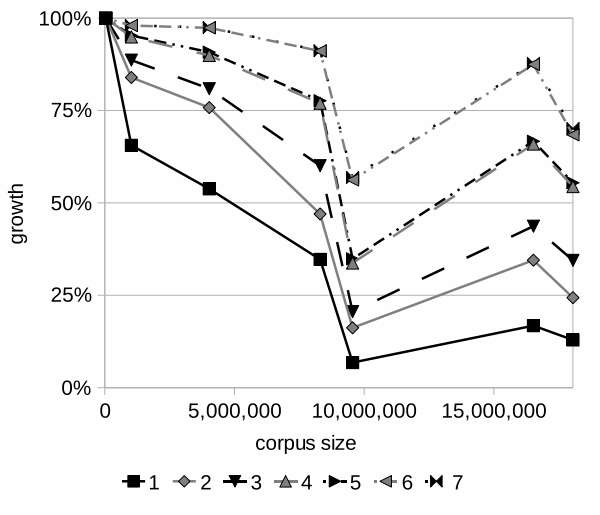
Although the results of the experiment suggest that we have reached some kind of ceiling in translation quality, we intend to at least double the size of the target language treebank and test whether we can break through these ceilings.

Figure 7 shows the percentages of new bags to be added to the database for each of the abstraction levels when gradually adding the subcorpora. Adding new corpora seems to add relatively little new information to the most abstract levels, but for the more concrete levels, growth percentages are still more than 50%, meaning that more than 50% of the bags found in the new corpus were unseen in the previous corpora.

We set up this experiment in order to estimate the upper bound of our MT system. Connecting this component to the other components of our MT system will reveal its true quality, but the results up to now are very encouraging.

We will also implement this approach for the other languages in our MT system, but probably with less abstraction levels. For instance, for English we use the Stanford parser (Klein and Manning, 2003), which generates parts-of-speech, dependency relations, categories, and words, but not frames or anything equivalent.

Figure 7: Growth percentage for each abstraction level



References

Bod, R. (1992). A Computational Model of Language Performance: Data-Oriented Parsing. In. C. Boîtet (ed.), *Proceedings of the fifteenth International Conference on Computational Linguistics (COLING'92)*. International Committee on Computational Linguistics. Nantes, France. pp. 855-859.

Carl, M., Melero, M., Badia, T., Vandeghinste, V., Dirix, P., Schuurman, I., Markantonatou, S., Sofianopoulos, S., Vassiliou, M., and Yannoutsou, O. (2008). METIS-II: Low Resources Machine Translation : Background, Implementation, Results, and Potentials. *Machine Translation* 22(1). pp. 69-99. Springer.

Chen, S.F., and Goodman, J. (1998). An Empirical Study of Smoothing Techniques for Language Modeling. *Technical Report TR-10-98*. Computer Science Group, Harvard U., Cambridge, MA.

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics. In *Proceedings of the Second Human Language Technology Conference (HLT)*. Morgan Kaufmann. San Diego, USA. pp. 138-145.

Hearne, M. (2005). *Data-Oriented Models of Parsing and Translation*. PhD thesis. Dublin City University. Ireland.

Klein, D., and Manning, C. (2003). Accurate Unlexicalized Parsing. In *Proceedings of 41st Annual Meeting of the Association of Computational Linguistics (ACL)*. Sapporo, Japan. pp. 423-430.

Koehn, P. (2005). Europarl. A parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*. Phuket, Thailand. pp. 79-97.

Lønning, J.T., Oepen, S., Beermann, D., Hellan, L., Carroll, J., Dyvik, H., Flickinger, D., Johannsen, J.B., Meurer, P., Nordgård, T., Rosén, V., and Velldal, E. (2004). LOGON. A Norwegian MT effort. In *Proceedings of the Workshop in Recent Advances in Scandinavian Machine Translation*. Uppsala, Sweden.

Macken, L., and Daelemans, W. (2009). Aligning linguistically motivated phrases. In *Computational Linguistics in the Netherlands 2007: Selected papers from the eighteenth CLIN meeting*. LOT Netherlands Graduate School of Linguistics. Utrecht. pp. 37-52

Och, F., and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29 (1), pp. 19-51.

Oostdijk, N., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.P., Moortgat, M., and Baayen, H. (2002). Experiences form the Spoken Dutch Corpus Project. In *Proceedings of the 3rd International conference on Language Resources and Evaluation (LREC)*. Las Palmas, Spain. pp. 340-347.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: a method for automatic evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. Philadelphia, USA. pp. 311-318.

Pollard, C., and Sag, I. (1994). *Head-driven Phrase Structure Grammar*. CSLI Stanford. University of Chicago Press. Stanford, USA.

Poutsma, A. (1998). Data-Oriented Translation. Presented at the *Ninth Conference of Computational Linguistics in the Netherlands*. Leuven, Belgium.

Snover, M., Dorr, B., Schwartz, R., Micciula, L, and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation inthe Americas (AMTA)*. Cambridge, USA. pp. 223-231.

Stolcke, A. (2002). SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*. Denver, Colorado, September 2002.

Tiedemann, J. (2003). *Recycling Translations - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. PhD. Studia Linguistica Upsaliensia 1.

Tinsley, J., Zechev, V., Hearne, M., and Way, A. (2007). Robust Language Pair-Independent Sub-Tree Alignment. *Proceedings of MT Summit XI*. Copenhagen. pp. 467-474.

Vandeghinste, V. (2007). Removing the Distinction Between a Translation Memory, a Bilingual Dictionary and a Parallel Corpus. In *Proceedings of Translating and the Computer*, 29. ASLIB. London, UK.

Vandeghinste, V. (2008). *A Hybrid Modular Machine Translation System*. PhD. Katholieke Universiteit Leuven. LOT Netherlands Graduate School of Linguistics. Utrecht.

van Noord, G., Schuurman, I., and Vandeghinste, V. (2006). Syntactic Annotation of Large Corpora in STEVIN. In *Proceedings of the 5th International conference on Language Resources and Evaluation LREC*. Genova, Italy.

van Noord, G. (2006). At Last Parsing Is Now Operational. In *Proceedings of TALN*, Leuven, Belgium.

Language Model Adaptation for Difficult to Translate Phrases

Behrang Mohit * **Frank Liberato** ** **Rebecca Hwa** *,**
Intelligent Systems Program *, Department of Computer Science **
University of Pittsburgh, Pittsburgh, PA 15260, USA
{behrang,liberato,hwa}@cs.pitt.edu

Abstract

This paper investigates the idea of adapting language models for phrases that have poor translation quality. We apply a selective adaptation criterion which uses a classifier to locate the most difficult phrase of each source language sentence. A special adapted language model is constructed for the highlighted phrase. Our adaptation heuristic uses lexical features of the phrase to locate the relevant parts of the parallel corpus for language model training. As we vary the experimental setup by changing the size of the SMT training data, our adaptation method consistently shows strong improvements over the baseline systems.

1 Introduction

Statistical Machine Translation (SMT) systems generally use the same setup to translate all sentences. During decoding, the SMT engine searches through a large set of model parameters. Many parameters are sparse, irrelevant and noisy with respect to the individual sentence that is being translated. The dominant solution to this problem is to train with a larger corpus. This addresses the data sparsity problem, but it creates more irrelevant model parameters. Moreover, large volumes of training data may not always be available.

In this paper, we consider ways of filtering the irrelevant and noisy parameters in order to improve translation quality. We propose a language model adaptation method for the translation of phrases. We construct one adapted language model per source language phrase by using its lexical features. Our method uses these features along with the parallel corpora sentence associations to locate the relevant target language training sentences. Furthermore, we examine the idea of adapting the language model based on the level of difficulty that a phrase presents to the SMT system. We estimate the translation difficulty of phrases in a pre-translation step with either gold standard labeling or a trained classifier. We find that only phrases that are deemed difficult for the SMT system, benefit from the adaptation. Finally, we asses the feasibility of our selective adaptation within a complete SMT pipeline.

2 Motivation

Previously we explored the estimation of the translation difficulty of phrases, using an automatic MT evaluation (BLEU) score (Mohit and Hwa, 2007). We used our estimation method to label a set of gold standard phrases with the difficulty information. Moreover, we showed that it is possible to automatically learn the translation difficulty by constructing a phrase difficulty classifier.

In this paper, our aim is to improve MT quality by focusing on what we call Difficult To Translate Phrases (DTPs) within source language sentences. We compiled a group of DTPs for a baseline SMT system. We then manually examined a group of difficult phrases to learn about the reasons that make these phrases difficult.

Among various, often overlapping reasons behind phrase difficulty, we frequently observed problems that can be reduced by modifying the language models. Not all language model problems are related to sparsity. Moreover, the focus of this paper is on problems that arise when the models are not sparse. Specifically we aim to address the following problems:

Proceedings of the 13th Annual Conference of the EAMT, pages 160–167,
Barcelona, May 2009

i. Disambiguating target language words: Target language word ambiguity can be reduced if there are distinct source language words. For example, the word *official* is ambiguous in English (person vs. feature), but it has two distinct Arabic translations for its two senses. An adapted language model trained in the right text domain, can filter in or out generation of a phrase like *Egyptian official* or *official Egyptian*.

ii. Short distance word movements: Language model can also help the decoder to decide about short term word movements. For example the ordering of adjective and nouns are reverse in Arabic and English. Generation of a phrase like *senior egyptian cleric* depends on the n-gram parameters associated with that phrase[1]. A language model that is fitted for generation of the above phrase, is likely to have a high trigram probability for the actual trigram or has high probability for the two bigrams: *senior egyptian* and *egyptian cleric* and lower probabilities for alternative bigrams such as cleric senior.

In order to solve the above problems, we bias the language model towards the domain of the translation task. Through this biasing, we filter out those parts of the training data that are irrelevant. The resulting language model is adapted for the translation of specific input (in our case, phrase). This approach is the opposite of the typical method of reducing model sparsity via data expansion. In other words, we deliberately create model sparsity in areas that are found irrelevant to the translation task.

3 Contributions

In this paper we implement methods and experiments to answer the following questions:

i. How do we adapt the language model to overcome the translation difficulties noted in Section 2?

ii. What is a reasonable upper bound estimate of quality improvements that can be gained from adapted language models?

iii. Do all phrases have an improvement in translation from model adaptation?

iv. In a general MT test, does our proposed model adaptation framework improve translation quality?

In the following we briefly explain our approach for answering each of these questions which will be followed by experimental details and results.

3.1 Our Adaptation Method

In SMT training, the target side of the parallel corpus is usually used for language model training. We would like to bias the language model training towards n-grams related to our translation phrase. To do so, we use the parallel corpus as a medium to locate relevant training instances. We start with the source language content words of the translation phrase. We call these source language terms, *seed* words. From the parallel corpus, we extract sentences that hold at least one of these seed words. We then include the associated target language sentence as one training sentence for the new language model. We call these training sentences, *relevant* sentences. This new relevant corpus is a much smaller subset of the original target language corpus. Some of the relevant sentences match longer n-grams with the translation task and we increase their training influence by repeating them. The repetition size is based on the length of the matched n-grams.

3.2 Estimating Upper Bounds

Estimating an upper bound for model adaptation gives us a realistic picture about the potentials of our constant resources (e.g, parallel corpus, etc.) and the expectations that we can have about language model adaptation. We present two methods which will more reliably gauge the impact of language modeling in the larger context of the decoder.

An aggressive upper bound: Given the constant phrase table, what is the closest possible decoding to the reference translation? To obtain such an upper-bound we simply train the language model with one reference translation. This ultra-overfited model is capable of generating sentences very close to the reference translation. However, the shortcomings of other translation resources such as unknown words or distortion errors are inherited when we use this language model. This upper bound tells us how much an n-gram language model, regardless of the training data, can be expected to improve translation quality.

A realistic upper bound: Unlike the aggressive upper bound scenario, in practice we train the language model on the target side of the parallel corpus. We are interested to estimate the best language model that we can build from that corpus. We still assume that we have access to the refer-

[1]For this example, we are assuming that the phrase table is only providing the word to word translations

ence translation, but we no longer include it directly in the training data. Instead, we assume that we have a mechanism to choose the relevant parts of the target language corpus to train a language model. In order to train this upper bound we follow these steps:

for each n-gram in the reference translation: **do**
  if n-gram holds a content word: **then**
    Pick training sentences that hold the n-gram
  end if
  Use n-gram size to weight each training sent.
end for

In our experiments we will compare the above two upper bounds estimates against the traditional method of expanding the data for the language model training.

3.3 Model Adaptation For Phrase Translation

We would like to know if the translation of all phrases can be improved by the model adaptation. To answer this question, we apply our model adaptation method to two phrase groups: Difficult and Easy to Translate. We extract and label sets of gold standard phrases based on our variation of the procedure explained in Section 2. These sets are sentences whose most difficult or easy phrase is highlighted as the *focus* phrase. Each focus phrase is translated as part of a sentence. However, we only adapt the language model for the translation of the focus phrase, while the rest of the sentence is translated with the baseline language model.

3.4 Model Adaptation within an MT framework

Finally we would like to know if model adaptation improves the translation quality of a complete MT pipeline. We employ our adaptation method as part of a pre-translation pipeline. We still apply the adaptation to the DTP part, however we use a phrase difficulty classifier to find the most difficult phrase of each sentence. As shown in Figure 1, after the classifier finds the difficult phrase, the adapted language model is constructed for it.

After the creation of the adapted language model, the phrase is translated in the context of the full sentence, similar to the previous section.

4 Experimental Setup

We conduct our experiments on translation of Arabic to English via a phrase-based statistical

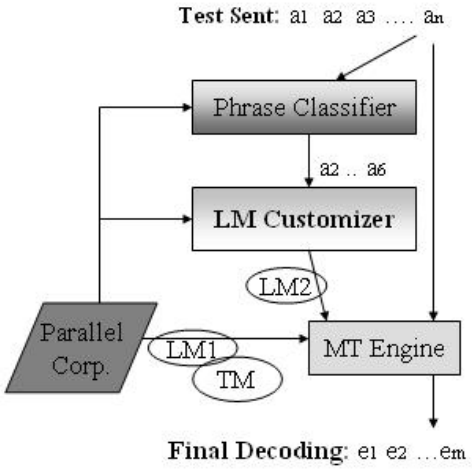


Figure 1: Translation Pipeline for Adapted LMs

translation engine. The SMT engine is the open source Phramer decoder (Olteanu, 2006) that uses the same training and decoding framework of the Pharaoh decoder (Koehn, 2004). We modify the decoder to use alternative language models for the translation of a special phrase within each sentence. We train both the baseline and the adapted language models using the SRI language modeling package (Stolcke, 2002).

4.1 Two SMT Systems

We construct two translation systems by varying the size of the training corpora. This data variation enable us to asses our approach in different translation scenarios. We use Arabic-English parallel corpora released by the Linguistic Data Consortium (LDC). The first (*small*) system is trained with one million words of parallel corpus[2]. The second *(medium)* system is cumulatively trained on an LDC corpus of 50 million words[3]. The language models for both systems are trained by the target language side of the parallel corpora. Both systems are tuned, using a development set of 500 sentences.

We use the LDC's multi-translation Arabic-English corpus [4] to extract a set of 3360 parallel phrases and label them as easy or difficult to translate. Since difficulty labels are system-specific, we label parallel phrases for both the small and medium systems. Phrase labeling and translation

[2]The corpora can be obtained from the Linguistic Data Consortium under catalog ID LDC2004T17, LDC2004T18.
[3]LDC2004E13, LDC2004E72, LDC2005E46
[4]LDC2003T18

evaluation are done based on the BLEU score (Papineni et. al., 2002).

4.2 Modified Easy-Difficult Phrase Labeling

Using the alignment-based phrase extraction tools, we automatically extract a corpus of parallel phrases. Each phrase consists of 5 to 15 source language tokens. The phrase set totally includes 32% of the sentences within the corpus. We use a held out parallel corpus to label each phrase as easy or difficult in the following round-robin fashion: Taking the translation BLEU score of the held out corpus as the referencee point, we add one phrase translation at a time and re-calculate the BLEU score. If the BLEU score is improved, the added phrase is an easy phrase (whose translation has improved the score). Otherwise, the phrase is labeled as difficult. In our labeling we use the first three reference translations to compute the BLEU scores. We keep the last reference for future MT quality evaluation. This separation reduces the bias of our labeling on our further experiments.

4.3 Building Difficulty Classifiers

When applying the language model adaptation to our full translation pipeline (Section 3.4), we use the phrase difficulty classifier to highlight the most difficult phrase of each sentence to replace our gold-standard labels. For each SMT system (small, medium), we construct a separate difficulty classifier. We compile a set of 12 language modeling features for the source and target languages to train these difficulty classifiers. We use the Support Vector Machine (SVM) as the classification model. These classifiers use the polynomial kernels which are tuned with a set of 100 development phrases.

4.4 Large Language Model Training

We also compare language model adaptation with the traditional method of using larger language models. We construct larger language models by adding up to 200 million words of the target language text to the language model training data.[5]. The larger language model's training data is cumulative with respect to the baseline models. These larger models are not in the scale of the current state of the art ultra-large models (Brant et. al., 2007). However by modifying the size of the parallel corpora along with the larger language models, we aim to simulate different data size scenarios. In this paper we report experiments where the larger language model is only used for translation of one phrase within the sentence (easy or difficult).

[5]This is a randomly chosen, continuous subset of the English Gigaword corpus.

5 Experiments

We conduct two sets of experiments by varying the accuracy of easy-difficult labeling. In the first setup (Sections 5.1 and 5.2), we use gold standard difficult or easy phrases with their associated sentences[6]. In the second setup (Section 5.3), we train an easy-difficult phrase classifier and use it to find the appropriate phrase. In both experiments, we modify the language model for the translation of the highlighted phrase and translate the sentence.

5.1 Model Modification for Difficult Phrases

We start with the upper bound estimates of language model adaptations. As explained in Section 3.2, there are two upper bound language models: the realistic upper bound and the aggressive upper bound. Table 1 presents the phrase level evaluation of these upper bound estimates along with the baseline system, the larger language models, and our adaptation method. Using the realistic upper bound language models, difficult phrases get sharp improvements. This large gap is indicative of the strong potentials that the baseline training data and the idea of language model adaptation hold. Table 2 presents a sentence level evaluation of the same experiments. Since the language model modifications are applied only to one difficult phrase per sentence, the score variations are smoothed at the sentence level. However, we still observe strong score improvements for the upper bounds and our adaptation method.

LM Modif.	Small Sys	Med Sys
Baseline	16.96	18.58
100M wds LM	21.17	21.29
200M wds LM	21.92	21.83
Realistic U.B.	26.83	28.33
Aggressive U.B.	54.23	60.11
Our Adapt.	21.12	22.16

Table 1: Comparison of different LM modifications for DTPs (Phrase Level Evaluation).

[6]For the small system we use a set of 453 easy and 551 difficult phrases. For the medium system, we use a set of 471 easy and 544 difficult phrases.

LM Modif.	Small Sys	Med Sys
Baseline	21.63	25.90
100M wds LM	23.76	27.04
200M wds LM	23.85	27.55
Realistic U.B.	25.98	30.07
Aggressive U.B.	35.92	37.46
Our Adapt.	23.47	27.93

Table 2: Comparison of different LM modifications for DTPs (Sentence Level Evaluation).

When we use our language model adaptation, each sentence is translated with language models that only use 5-10% of the baseline training data. Both systems' results indicate strong improvements above the baseline system and competitive performance with the large language models. For example the results of model adaptation based on a corpus of one million words competes with the results of models trained on corpora in the scales of 100 or 200 million words.

The improvements from our adaptation method relate to the way that language model influences DTPs. We have observed that the phrase table tends to be sparse for DTPs. As a result, the DTPs are often translated word for word so that the language model has to compensate for the word order. Our adaptation method is aimed at sharpening the discriminative power of the relevant n-grams. By filtering out the irrelevant training data, we distribute the probability mass only among the n-grams that are relevant to the translation phrase. The resulting small adapted language model is tailored for discriminating between a special set of n-grams that are relevant to our translation task.

5.2 Should we modify the model for all phrases?

In the above experiments, we applied language model modifications to phrases that are difficult to translate, and observed improvements in translation quality. However, it is not clear if this pattern of improvements applies to all phrases. To find the answer, we repeat those experiments for easy to translate phrases. Tables 3 and 4 present the result of these experiments. Contrary to difficult phrases, easy phrases which are completely tuned towards the baseline language model do not gain strong improvements. In some experiments their translation quality actually deteriorates.

For example easy phrases gain a modest improvement from the realistic upper bound language model. This indicates how close the baseline language model is to our (approximately) ideal language model. In other words, the parameters of the baseline language model are tuned towards generation of sentences close to the easy phrase references. The easy phrases are so fine tuned with the baseline language model that even a much larger language model can not compete with the baseline language model.

LM Modif.	Small Sys	Med Sys
Baseline	41.81	45.45
200M wds LM	39.26	44.03
Realistic U.B.	42.91	47.19
Aggressive U.B.	73.17	74.05
Our Adapt.	41.77	45.09

Table 3: Comparison of different LM modifications for easy phrs (Phrase Level Evaluation).

LM Modif.	Small Sys	Med Sys
Baseline	27.75	32.70
200M wds LM	26.93	32.02
Realistic U.B.	28.12	33.66
Aggressive U.B.	37.19	39.93
Our Adapt.	27.64	32.57

Table 4: Comparison of different LM modifications for easy phrs (Sentence Level Evaluation).

The results for model modification of easy phrases show that model adaptation can be more effective if it is applied selectively to only difficult phrases. In order to apply selective model adaptation, we need a mechanism to find those difficult phrases that need special handling. Therefore we construct a translation difficulty classifier.

5.3 Selective Model Adaptation for SMT

In this experiment, we apply our model adaptation into a complete SMT pipeline. Here, we use the Figure 1 architecture to find the most difficult phrase of a sentence and selectively modify the language model. For translation of each sentence, we apply the following procedure:

i. Compile the set of all source language phrases. To reduce the scale and keep the procedure similar to our gold standard labeling, we only consider phrases that have 5 to 15 words and have a contiguous baseline translation.

ii. Extract classification features for all phrases

LM Modif.	Small Sys	Med Sys
Baseline	18.09	22.51
Our Adapt.	19.06	23.55

Table 5: An Start-to-Finish experiment with difficulty classifier in the SMT pipeline

and their translations.

iii. Classify all phrases of a sentence and use the classifier's score to choose the most difficult phrase.

iv. Construct the modified language model for the most difficult phrase.

v. Translate the sentence by using the modified language model for the difficult phrase and the baseline setup for the rest of the sentence.

For this experiment, we use a held out Arabic-English test set[7]. For both small and medium sized systems, we experiment with the baseline language model and using our model adaptation method. Table 5 compares these three variations with the baseline.

For both systems, there are steady quality improvements above the baseline. It is clear the improvements are not as strong as the case where we apply the adaptation to gold standard DTPs. This is due to classification errors where a difficult phrase is missed or an easy phrase might be classified as difficult and gets selected for model adaptation where the new model might deteriorates the phrase's translation.

6 Discussion

We worked on the problem of modifying the language model to improve translation quality of difficult phrases. From various experiments we have observed the following:

i. Language modeling plays a significant role in SMT and strongly influences the difficulty of translation.

ii. A selective adaptation of the model based on the characteristics of the translation task has a strong potential to explore.

iii. Parallel data can be heuristically used to adapt language models based on the translation task.

We modified the baseline language models in two ways: we filtered out the irrelevant training data, and highlighted the relevant part of the remaining data based on n-gram matches. In the filtering part, we aim at removing some of the target word senses that are irrelevant to the translation task. We should clarify that here we do not address the problem of data sparseness. We actually cut some portions of the (irrelevant) baseline data. However, our filtering along with the appropriate weight setting, modify the relevant parameters of the model and make them biased towards the proper domain.

Table 6 presents sample translations where language model adaptation improves translation quality. The improvement in the first sample is related to the filtering aspect of the model adaptation. The English word *official* has two senses that are mixed up in the baseline translation. Since on the Arabic side the word has two distinct meanings, given the translation phrase and the parallel corpus we are able to exclude or lower the weight of the English sentences that have the irrelevant sense. The second sample is related to case where there is trigram match (egyptian police officer) between the parallel corpus and the translation phrase. The adaptation method locates the relevant sentence and increases its weight in the new language model training.

Our adaptation method is presented as an alternative to employing a larger amount of training data. In Figure 2, we compare various sizes of language models with the language model adaptation for difficult phrases. For both small and medium systems, our proposed adaptation method is competitive with the use of larger language models. Moreover, the realistic upper bound that uses the baseline training data is well above the largest tested language model. This encourages the further exploration of our idea.

A comparison of Tables 1 and 3 shows that the large quality gap between the easy and difficult to translate phrases holds even when we use the aggressive upper bound language models. This large gap shows the limits of influence for language models, especially for the difficult phrases. Due to numerous overlapping reasons, the language model can not solely resolve all the difficulties of the translation, so the translation quality of the difficult phrases stands well bellow the easy phrases.

[7]LDC2005T05: 606 sentences

baseline LM: the first exhibition for egyptian official of the painting on the UNK *Adapted LM:* the first official egyptian exhibition for the painting on the UNK *Reference:* the first official egyptian exhibition for painting on porcelain
baseline LM: ... that mohamed atef and is police officer former egyptian ... *Adapted LM:* ... that mohamed atef , is one of former egyptian police officer... *Reference:* ... that mohamed atif , a former egyptian police officer ...

Table 6: Sample Translation Improvements

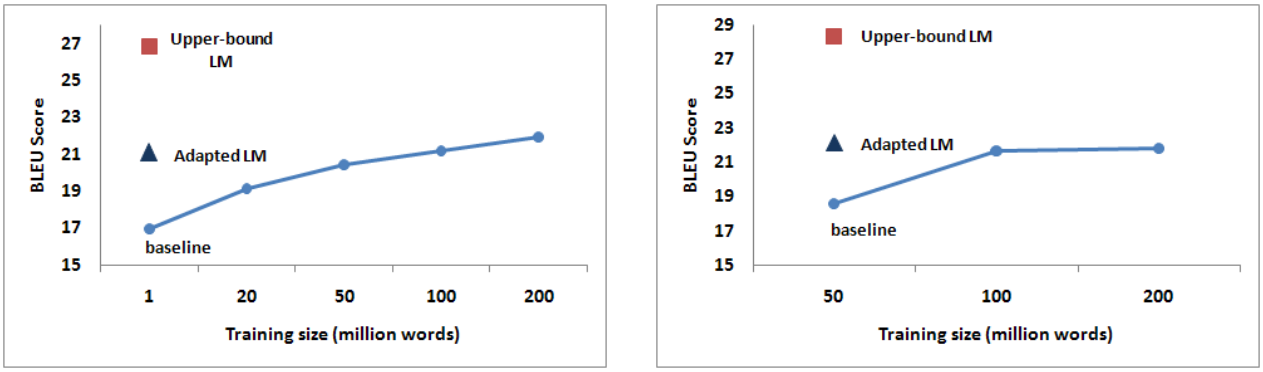


Figure 2: A comparison of model adaptation and training data expansion for systems that are trained on Small (Left) and Medium (Right) size parallel corpora

7 Related Work

Language Model Adaptation has been studied extensively in the speech processing and SMT communities (Kim and Khudanpur, 2004), (Tam et. al., 2007), (Snover et. al., 2008). The training data selection in most of the previous works involves selection of longer chunks of text. In contrast, we select training data for translation of each individual sentence. Also model adaptation has been mainly used in the translation of the entire test corpus with no special condition. Instead, we selectively use the adaptation only for translation of difficult phrases.

Our work is about improving the translation of phrases in the context of the sentence. Koehn and Knight (2003) present a frame work of isolated translation of noun phrases and re-combining the phrase translation with the rest of the sentence. Our phrase translation takes place in the context of the entire sentence but with a different language model. We still benefit from the full sentence context which reduces the translation error. The idea of decomposing the translation sentence and reattaching phrasal decodings has been also studied in the Multi Engine MT (MEMT) community. Mellebeek et. al. (2006) choose syntactically meaningful segments to decompose the sentences. In our approach we do not consider any syntactic constraint. Our major constraint is the translation difficulty of the phrase which makes us choose an alternative translation framework.

Another relevant area of work is training data subsampling. Johnson et. al. (2007) use the Fisher's exact test to validate the accuracy of the phrase table entries. They are able to reduce the size of the phrase table to 10% of its original size without a major loss of translation quality. It is not clear what percentage of the original training data is required to construct the reduced translation model. However their work confirms that training data can be used more efficiently. Ueffing et. al. (2007) also applies transduction learning to bootstrap new training sentences for SMT. New source language sentences are translated via a baseline SMT engine. Confidence estimation and model parameters are used for deciding to keep the sentence (and its decodings) in the new round of training. Our work follows a similar idea for altering

the training data, but instead of adding additional data, we filter out and reweigh training data based on the relevancy to the translation task.

Comparison of different language models in SMT is one of our challenges. In this paper, we used the end result (translation) quality to evaluate the language models. An alternative method to consider is the *gold-in-sands* framework (Zhang, 2008). The idea is that a better language model should be able to rank the reference translations higher than alternative translations. It is implied that such a language model is more capable of generating sentences close to the reference translation.

8 Conclusion

In this paper, we presented a heuristics for training language models that are adapted for specific phrases deemed difficult to translate. When applied to a complete SMT pipeline, our model adaptation method improves the translation quality and competes with larger language models. For many target languages where large volumes of monolingual data is not available, usage of a larger language model is not an option and model adaptation is even more helpful.

We are working in several directions: We would like to extend our model adaptation by considering the interaction between the translation and the language models. Also, we are interested to expand our comparisons of the adapted language models vs. other language models (eg. baseline), outside the MT decoding. One area to explore is the usage of language model-based re-ranking of the reference translations.

Acknowledgments

This work has been supported in parts by US National Science Foundation Grants IIS-0832381 and IIS-0745914. We would like to thank our reviewers, the NLP group at the University of Pittsburgh and the MT group at the Carnegie Mellon University for their valuable comments and suggestions.

References

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, Jeffrey Dean 2007. Large Language Models in Machine Translation In Proceedings of *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning(EMNLP-CoNLL-07)*.

J Howard Johnson, Joel Martin, George Foster and Roland Kuhn 2007. Improving Translation Quality by Discarding most of Phrase Table In Proceedings of *EMNLP-CoNLL-2007*.

W. Kim and S. Khudanpur. Cross lingual latent semantic analysis for language model. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the International Conferences on Acoustic Speech and Signal Processing, ICASSP 2004*.

Philip Koehn and Kevin Knight 2003. Feature-Rich Statistical Translation of Noun Phrases In Proceedings of *43rd Annual Meeting of Association for Computational Linguistics (ACL-03)*.

Philip Koehn 2004. Pharaoh: A Beam Search Decoder for Phrase-based Statistical Machine Translation Models Technical Report, USC Information Sciences Institute.

Bart Mellebeek, Karolina Owczarzak, Josef Van Genabith and Andy Way. 2006. Multi-Engine Machine Translation by Recursive Sentence Decomposition In Proceedings of *American Machine Translation Association Conference (AMTA-06)*.

Behrang Mohit and Rebecca Hwa 2007. Localization of Difficult-to-Translate Phrases. In Proceedings of *ACL Workshop on Statistical Machine Translation*.

Marian Olteanu 2006. Phramer: An open-source statistical phrase-based mt decoder In Proceedings of *NAACL Workshop on Statistical Machine Translation*.

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 Association for Computational Linguistics (ACL-2002)*.

Matthew Snover, Bonnie Dorr and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora In Proceedings of *Conference on Empirical Methods in Natural Language Processing (2008)*.

Andreas Stolcke 2008. SRILM – an extensible language modeling toolkit. In Proceedings of *Conference on Empirical Methods in Natural Language Processing (2002)*.

Yik-Cheung Tam, Ian Lane, and Tania Schultz. Bilingual-lsa based lm adaptation for spoken language translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Nicola Ueffing, Gholamreza Haffari and Anoop Sarkar 2007. Transductive Learning for Statistical Machine Translation In Proceedings of *EMNLP-CoNLL-2007*.

Ying Zhang 2008. Structured Language Model for Statistical Machine Translation PhD Thesis, Carnegie Mellon University. Pittsburgh, USA.

A Phrase-Based Hidden Semi-Markov Approach to Machine Translation

Jesús Andrés-Ferrer
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
jandres@iti.upv.es

Alfons Juan
Dpto. de sist. informáticos y computación
Universidad Politécnica de Valencia
ajuan@dsic.upv.es

Abstract

Statistically estimated phrase-based models promised to further the state-of-the-art, however, several works reported a performance decrease with respect to heuristically estimated phrase-based models. In this work we present a latent variable phrase-based translation model inspired by the hidden semi-Markov models, that does not degrade the system. Experimental results report an improvement over the baseline. Additionally, it is observed that both Baum-Welch and Viterbi trainings obtain the very same result, suggesting that most of the probability mass is gathered into one single bilingual segmentation.

1 Introduction

The machine translation problem is stated as the problem of translating a *source* sentence, $\boldsymbol{x}_1^J$, into a *target* sentence, $\boldsymbol{y}_1^I$. In accordance with the statistical approach to machine translation, the optimal translation $\hat{\boldsymbol{y}}$ of a source sentence $\boldsymbol{x}$ is given by the fundamental equation of statistical machine translation (Brown and others, 1993)

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y} \in \mathcal{Y}^\star} p(\boldsymbol{x} \mid \boldsymbol{y})\, p(\boldsymbol{y}) \qquad (1)$$

where $p(\boldsymbol{x} \mid \boldsymbol{y})$ is approximated by an *inverse translation model* and *p(**y**)* is modelled with a *language model*; which is usually instanced by a *n-gram language model* (Chen and Goodman, 1996).

The first approaches to model the translation probability in Eq. (1), were based on word dictionaries. These word-based models, the so-called *IBM translation models* (Brown and others, 1993), tackled the problem with word-level dictionaries plus alignments between words. However, current systems model the inverse conditional probability in Eq. (1) using *phrase dictionaries*. A phrase is understood here as any sequence of source or target words. This phrase-based methodology stores specific sequences of target words (*target phrase*) into which a sequence of source words (*source phrase*) is translated.

However, a key concept of this approach is the procedure through which these phrase pairs are inferred. The common approach consists in using the IBM alignment models (Brown and others, 1993) to obtain a symmetrised alignment matrix from which *coherent* phrases are extracted (Och and Ney, 2004). Then, a simple count normalisation is carried out in order to obtain a conditional phrase dictionary.

Alternatively, some approaches infer the phrase dictionaries statistically. For instance, a joint probability model for phrase-based estimation is proposed in (Marcu and Wong, 2002). In that work, all possible segmentations were extracted using the EM algorithm (Dempster et al., 1977), without any matrix alignment constraint, in contrast to the approach followed in (Och and Ney, 2004). Based on this work, another work (Alexandra Birch and Koehn, 2006), constrained the EM to only consider phrases which agree with the alignment matrix, thus reducing the size of the phrase dictionaries (or tables).

A possible drawback of the above phrase-based models is that they are not conditional, but joint models that require a re-normalisation post-processing in order to obtain a conditional model. However, a generative conditional phrase-based model presented in (DeNero et al., 2006) showed a worsening of phrase dictionaries.

Proceedings of the 13th Annual Conference of the EAMT, pages 168–175,
Barcelona, May 2009

In this work, we propose a conditional *phrase-based hidden semi-Markov model (PBHSMM)* that improves the phrase-dictionary estimation. Although, the improvements are not impressive, bare in mind that the main property of this model is its clear theoretical foundation, since it is based on a well-known statistical modelling technique, the so-called HSMM (Ostendorf et al., 1996). This allow us to include several statistical improvements into future revisions of the model (see section 5). A previous work (Andrés-Ferrer and Juan-Císcar, 2007) already presented a conditional phrase-based hidden Markov model (HMM). However our model presents significant improvements, both in theory and practice.

The model is detailed in section 2, while its EM-based training algorithms are analysed in section 3. Experiments are reported in section 4. Finally, concluding remarks are gathered in section 5.

2 The model

In this section, we present our *phrase-based hidden semi-Markov model (PBHSMM)* for machine translation. Hidden semi-Markov models (HSMMs) (Ostendorf et al., 1996) are a variation on HMM that allow the emission of segments $\boldsymbol{x}_j^{j+l-1}$ at each state instead of constraining the emission to one element x_j as HMM do. Therefore, the probability of emitting an object sequence $\boldsymbol{x}_j^{j+l-1}$ in any state depends on the segment length l. Note that in hidden Markov models (HMMs), the probability of emitting a segment of length l staying in the same state q, can only be simulated by transitions to the same state q. This yields the exponential decaying length probability expressed as follows

$$\mathrm{p}(l \,|\, q) = [\mathrm{p}(q \,|\, q)]^{l-1} \quad , \tag{2}$$

which is not appropriate for many situations.

The HSMM model introduced in this section is clearly inspired in the phrase-based translation models (Koehn et al., 2003). The idea behind this model is to provided a well-defined monotonic formalism that, while remaining close to the phrase-based models, explicitly introduces the statistical dependencies needed to define a phrased monotonic translation process. Although the monotonic constraint is an obvious disadvantage for this primer HSMM translation model, it can be extended to non-monotonic processes. However, these extensions lay far beyond the aim of this work.

Albeit there are several ways to formalise a HSMM, we advocate for a similar formalisation of that found in (Murhpy, 2007). Let $\boldsymbol{x} \in \mathcal{X}^\star$ be the source sentence and $\boldsymbol{y} \in \mathcal{Y}^\star$ the target sentence, then we start by decomposing the conditional translation probability, $p(\boldsymbol{x} \,|\, \boldsymbol{y}, I, J)$. We assume that the monotonic translation process has been carried out from left to right in sequences of words or *phrases*. For this purpose, both sentences should be segmented into the same amount of phrases. Figure 1, depicts an example of a possible monotonic bilingual segmentation in which the source sentence has a length of 9 words, while the target sentence is made up of 11 words. Note that each bilingual phrase makes up a *concept*; for instance c_1, c_2, c_3 and c_4 are concepts in Figure 1. To represent the segmentation process, we use two segmentation variables for both source, $\boldsymbol{l}$, and target, $\boldsymbol{m}$, sentences.

The target segmentation variable $\boldsymbol{m}$ stores each target segment length at the position at which the segment begins. Therefore, if the target segment length variable $\boldsymbol{m}$ has a value greater than 0 at position i, then a segment with length m_i starts at this position i. For instance, the target segmentation represented in Figure 1 is given by $\boldsymbol{m} = \boldsymbol{m}_1^{11} = (3,0,0,3,0,0,2,0,3,0,0)$. Note that values for the segment length variable such as, $\boldsymbol{m} = (3,0,0,3,0,0,2,0,1,0,0)$ or $\boldsymbol{m} = (3,0,0,3,0,0,1,0,3,0,0)$, *are invalid*. It is also worth noting that the domain of the segmentation ranges among all the possible segmentation lengths.

The source segmentation variable $\boldsymbol{l}$ represents the length of each *source segment* at the position at which its corresponding *target segment* ends. If the source segment length variable $\boldsymbol{l}$ has a value greater than 0 at position i; then the length of the source segment corresponding to the target segment that starts at position i, is l_i. For instance, in Figure 1 the source segment length variable is $\boldsymbol{l} = \boldsymbol{l}_1^{11} = (3,0,0,2,0,0,3,0,1,0,0)$.

Given a target segmentation variable, say $\boldsymbol{m}$, we define its prefix counterpart, $\bar{\boldsymbol{m}}$ as follows

$$\bar{m}_i = \sum_{k=1}^{i} m_k \quad i = 0,1,\ldots,I \quad . \tag{3}$$

In Figure 1, the prefix segments lengths are $\bar{\boldsymbol{m}}_0^{11} = (0,3,3,3,6,6,6,8,8,11,11,11)$ and

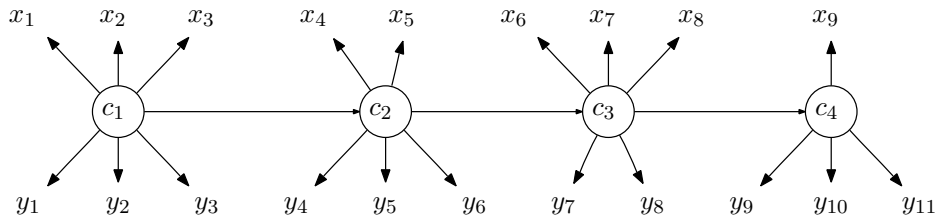


Figure 1: A generative example of the phrase-based hidden semi-Markov model for machine translation.

$\bar{\boldsymbol{l}}_0^{11} = (0, 3, 3, 3, 5, 5, 5, 8, 8, 9, 9, 9)$, for target and source segment length variables respectively.

Mathematically, we express the idea depicted in Figure 1 unhiding the former segmentation length variables

$$p(\boldsymbol{x} \,|\, \boldsymbol{y}) = \sum_{\boldsymbol{l}} \sum_{\boldsymbol{m}} p(\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{m} \,|\, \boldsymbol{y}, I, J) \quad . \tag{4}$$

The completed model in Eq. (4) is decomposed as follows

$$p(\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{m} \,|\, \boldsymbol{y}) := p(\boldsymbol{m}) p(\boldsymbol{l} \,|\, \boldsymbol{m}) p(\boldsymbol{x} \,|\, \boldsymbol{m}, \boldsymbol{l}, \boldsymbol{y}) \tag{5}$$

where we have dropped the dependence on $\boldsymbol{y}$ for the segment variables. Note that for clarity we have omitted the dependency on the lengths J and I in all probabilities; and we will henceforth proceed this way.

Both length probabilities in Eq. (5) are being decomposed left-to-right. However, in order to keep the training as fast as possible, a special decomposition of such probabilities is going to be made. We detail here the decomposition of the target segment length probability model, omitting details for the remaining random variables.

The probability of the target segment length variable is given by

$$p(\boldsymbol{m}) = \prod_{i=1}^{I} p(m_i \,|\, \boldsymbol{m}_1^{i-1}) \quad . \tag{6}$$

At first stage, we had assumed that each partial probability in Eq. (6) does not depend neither on $\boldsymbol{y}$, nor on both lengths (I and J). Hence, the probability $p(m_i \,|\, \boldsymbol{m}_1^{i-1})$ is modelled as follows

$$p(m_i \,|\, \boldsymbol{m}_1^{i-1}) = \begin{cases} \mathrm{p}(m_i) & \bar{m}_{i-1} + 1 = i,\ m_i \neq 0 \\ 1 & \bar{m}_{i-1} + 1 \neq i,\ m_i = 0 \end{cases} \tag{7}$$

Finally the segment length probability is expressed as follows

$$p(\boldsymbol{m}) := \prod_{i \in \mathcal{Z}(\boldsymbol{m})} 1 \prod_{i \notin \mathcal{Z}(\boldsymbol{m})} \mathrm{p}(m_i) \quad , \tag{8}$$

where $\mathcal{Z}(\boldsymbol{m})$ or simply $\mathcal{Z}$ stands for the set of positions t for which m_t is 0. For instance, in the example in Figure 1, $\mathcal{Z}$ is instanced to $\mathcal{Z}(\boldsymbol{m}) = \{2, 3, 5, 6, 8, 10, 11\}$.

Provided that one of the two products in Eq. (8) simplifies to 1, the segment length probability is expressed as

$$p(\boldsymbol{m}) := \prod_{i \notin \mathcal{Z}} \mathrm{p}(m_i) \quad . \tag{9}$$

Since explicitly showing these details forces the discourse to be awkward, we will omit these details. Therefore, we will use equations resembling the following

$$p(\boldsymbol{m}) := \prod_{t} \mathrm{p}(m_t) \quad , \tag{10}$$

where we have explicitly ommitted that $t \in \mathcal{Z}$, and we have changed the index i into t for subtly summarising the whole previous simplification process. This approach resembles the state probability decomposition in HSMM (Ostendorf et al., 1996).

Similarly to the target segment length model, the source segment length yields the following decomposition

$$p(\boldsymbol{l} \,|\, \boldsymbol{m}) := \prod_{t} \mathrm{p}(l_t \,|\, m_t) \quad . \tag{11}$$

Finally, knowing the length segment variables, the emission probability is also decomposed left-to-right as follows

$$p(\boldsymbol{x} \,|\, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{y}) := \prod_{t} \mathrm{p}(\boldsymbol{x}(t) \,|\, \boldsymbol{y}(t)) \quad , \tag{12}$$

where $\boldsymbol{y}(t)$ stands for $\boldsymbol{y}_t^{t+m_t-1}$ and $\boldsymbol{x}(t)$ stands for $\boldsymbol{x}_{\bar{l}_{t-1}+1}^{\bar{l}_t}$; i.e., the t-th "emitted" phrase and its respective t-th target phrase. Note that since t is a boundary of a target segment, then $\bar{l}_t$ is equal to $\bar{l}_{t-1} + l_t$.

Summarising, the proposed (completed) conditional translation model is defined by

$$p(\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{m} \,|\, \boldsymbol{y}) := \prod_t \mathrm{p}(m_t)\, \mathrm{p}(l_t \,|\, m_t)\, \mathrm{p}(\boldsymbol{x}(t) \,|\, \boldsymbol{y}(t)) \tag{13}$$

Then, the incomplete model introduced in Eq. (4) is parameterised as follows

$$p(\boldsymbol{x}|\boldsymbol{y}) := \sum_{\boldsymbol{l}} \sum_{\boldsymbol{m}} \prod_t \mathrm{p}(m_t)\, \mathrm{p}(l_t \,|\, m_t)\, \mathrm{p}(\boldsymbol{x}(t)|\boldsymbol{y}(t)) \tag{14}$$

with the following parameter set $\boldsymbol{\theta}$

$$\boldsymbol{\theta} = \{\mathrm{p}(m), \mathrm{p}(l \,|\, m), \mathrm{p}(\boldsymbol{u} \,|\, \boldsymbol{v})\} \tag{15}$$

where l and m are positive integers, $\boldsymbol{u}$ is a source phrase, i.e., $\boldsymbol{u} \in \mathcal{X}^\star$; and $\boldsymbol{v}$ is a target phrase $\boldsymbol{v} \in \mathcal{Y}^\star$.

It is important to smooth the phrase translation probabilities to avoid over-training. For doing so, we have used the IBM model 1 (Brown and others, 1993) as follows

$$\tilde{\mathrm{p}}(\boldsymbol{u} \,|\, \boldsymbol{v}) = (1-\epsilon)\ \mathrm{p}(\boldsymbol{u} \,|\, \boldsymbol{v}) + \epsilon\, p_{IBM1}(\boldsymbol{u} \,|\, \boldsymbol{v}) \tag{16}$$

Note that in this model, each target phrase $\boldsymbol{y}(t)$ is understood as the "state" of a HSMM in which the source phrase $\boldsymbol{x}(t)$ is emitted. Obviously this is not a pure HSMM in which we have a latent state variable. The omission of this latent variable is more an assumption than a requirement. Recall that in Figure 1 we have depicted each bilingual phrase pair being emitted by a *concept*. Therefore, we could theoretically model this latent variable as well. This inclusion would not significantly change the algorithms proposed here. However, this idea is left as future work, since it is firstly needed to check whether this primer model degrades or not the system performance as some similar works have previously reported (DeNero et al., 2006; Marcu and Wong, 2002).

3 The training

Since the proposed PBHSMM assumes that the segment length variables are not given in the training data, some approximate inference algorithm such as the EM (Dempster et al., 1977) is needed. We omit here the EM derivations which lead to the well-known Baum-Welch algorithm (Rabiner, 1990). This algorithm follows the iterative scheme of all the EM instantiations. First, we guess an adequate parameter set, $\boldsymbol{\theta}^{(0)}$, as a start point. Then, we compute the forward, $\alpha_{tl}^{(0)}(\boldsymbol{x}, \boldsymbol{y})$, and backward, $\beta_{tl}^{(0)}(\boldsymbol{x}, \boldsymbol{y})$, recurrences for each sample. These recurrences are used to compute the fractional counts $\gamma_{tlt'l'}^{(0)}(\boldsymbol{x}, \boldsymbol{y})$; and afterwards, a new $\boldsymbol{\theta}^{(1)}$ is estimated from those fractional counts. The re-estimated parameter set $\boldsymbol{\theta}^{(1)}$ can be used again to re-compute the recurrences, defining an iterative process that ensures the log-likelihood to increase in each iteration (or remain the same). This process goes on until either convergence or a maximum number of iterations is achieved.

3.1 Forward recurrence

The forward recurrence α_{tl} is defined as the prefix probability

$$\alpha_{tl} = \alpha_{tl}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_1^l, \bar{l}_t = l, \bar{m}_t = t \,|\, \boldsymbol{y}) \tag{17}$$

where $\bar{l}_t = l$ and $\bar{m}_t = t$ mean that a source and a target phrase end/start at position l of the input and t of the output. This prefix probability is recursively computed as follows

$$\alpha_{tl} = \begin{cases} 1 & t = 0, l = 0 \\ \sum_{t'} \sum_{l'} \alpha_{t'l'}\, \mathrm{p}(l'-l, t'-t) & 0 < t \le I, \\ \qquad \cdot\, \mathrm{p}(\boldsymbol{x}_{l'+1}^{l} | \boldsymbol{y}_{t'+1}^{t}) & 0 < l \le J \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where the sum over t' ranges from 0 to $t-1$ and likewise the sum over l' ranges from 0 to $l-1$; and where we have used $\mathrm{p}(l'-l, t'-t)$ to denote the product of lengths

$$\mathrm{p}(l'-l, t'-t) = \mathrm{p}(t'-t)\, \mathrm{p}(l'-l \,|\, t'-t) \quad , \tag{19}$$

in order to compress notation.

3.2 Backward recurrence

The backward recurrence β_{tl} is defined as the following suffix probability

$$\beta_{tl} = \beta_{tl}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{l+1}^{J} | \bar{l}_t = l, \bar{m}_t = t, \boldsymbol{y}) \tag{20}$$

where $\bar{l}_t = l$ and $\bar{m}_t = t$ mean that a source and a target phrase ended/started at position l of the input and t of the output. The former suffix probability is recursively computed as follows

$$\beta_{tl} = \begin{cases} 1 & t = I, l = J \\ \sum_{t'} \sum_{l'} \beta_{t'l'}\, \mathrm{p}(l'-l, t'-t) & 0 \le t < I, \\ \qquad \cdot\, \mathrm{p}(\boldsymbol{x}_{l+1}^{l'} \,|\, \boldsymbol{y}_{t+1}^{t'}) & 0 \le l < J \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where the sum over t' ranges from $t+1$ to I and likewise the sum over l' ranges from $l+1$ to J.

These two recurrences answer the question of which is the probability for a given pair of sentences

$$\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x} \,|\, \boldsymbol{y}) = \alpha_{IJ} = \beta_{00} \quad . \tag{22}$$

Both the forward and backward recurrence require a matrix of size $O(IJ)$. In order to compute these recurrences a time complexity of $O(I^2J^2)$ is required. However, it can be reduced to $O(IJM^2)$ by defining a maximum phrase length M.

3.3 Fractional counts

Using the previously defined recursions, we can compute the probability of segmenting a given sample in the source positions (l, l') and in the target positions (t, t')

$$\gamma_{tlt'l'} = \frac{\alpha_{tl}\,\mathrm{p}(l'-l, t'-t)\,\mathrm{p}(\boldsymbol{x}_{l+1}^{l'} \,|\, \boldsymbol{y}_{t+1}^{t'})\beta_{t'l'}}{\mathrm{p}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y})} \tag{23}$$

This fractional count is very helpful through the Baum-Welch training.

3.4 Re-estimation

Once we have computed the recurrences and the fractional counts, the phrase translation probabilities are re-estimated as follows

$$\mathrm{p}(\boldsymbol{u} \,|\, \boldsymbol{v}) = \frac{N(\boldsymbol{u}, \boldsymbol{v})}{\sum_{\boldsymbol{u}'} N(\boldsymbol{u}', \boldsymbol{v})} \tag{24}$$

with

$$N(\boldsymbol{u}, \boldsymbol{v}) = \sum_n \sum_{l<l'} \sum_{t<t'} \gamma_{ntlt'l'} \delta(\boldsymbol{x}_{l+1}^{l'}, \boldsymbol{u}) \delta(\boldsymbol{y}_{t+1}^{t'}, \boldsymbol{v}) \tag{25}$$

where $\delta(a, b)$ is the Kronecker delta function which is 1 if $a = b$ and 0 otherwise.

The target phrase length probabilities are estimated as follows

$$\mathrm{p}(m) = \frac{N(m)}{\sum_{m'} N(m')} \tag{26}$$

with

$$N(m) = \sum_n \sum_{l<l'} \sum_t \gamma_{n,t,l,(t+m),l'} \tag{27}$$

Finally, the source phrase length probabilities are re-estimated by

$$\mathrm{p}(l \,|\, m) = \frac{N(l, m)}{\sum_{l'} N(l', m)} \tag{28}$$

with

$$N(l, m) = \sum_n \sum_{l'} \sum_t \gamma_{n,t,l,(t+m),(l'+l)} \tag{29}$$

where l denotes a source phrase length, and m a target phrase length.

An alternative training algorithm is obtained computing the maximum segmentation instead of the recurrences. This training, the so-called Viterbi training (Rabiner, 1990), is an iterative training process as well. Each iteration comprises two stages: computing the maximum segmentation and re-estimating the parameters. The Viterbi recursion is used to obtain the maximum segmentation

$$\delta_{tl} = \begin{cases} 1 & t = 0, l = 0 \\ \max_{t',l'} \{\delta_{t'l'}\,\mathrm{p}(l'-l, t'-t) & 0 < t \leq I, \\ \qquad \mathrm{p}(\boldsymbol{x}_{l'+1}^{l} | \boldsymbol{y}_{t'+1}^{t})\} & 0 < l \leq J \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

A traceback of the decisions made to compute δ_{IJ} provides the maximum segmentation $\hat{\boldsymbol{m}}$ and $\hat{\boldsymbol{l}}$.

Afterwards, the re-estimation equations are the similar to Eqs. (24), (26), and (28), but in this case the counts $N(\boldsymbol{u}, \boldsymbol{u})$, $N(m)$, and $N(l, m)$ are the actual counts since the latent segmentation is assumed to be the maximum segmentation.

4 Experiments

The aim of the experimentation is to see how the proposed method and algorithm improves the quality of a any phrase dictionary given as input. For doing so, we have tested our algorithm in two corpora: the Europarl-10 and the Europarl-20. The former comprises all the sentences from the English-to-Spanish part of Europarl (version 3) (Koehn, 2005) with length equal or less than 10. The latter is made up of all the English-to-Spanish Europarl sentences with length equal or less than 20. For both corpora we have randomly selected 5000 sentences for testing the algorithms. Note that we have constrained the training length of the standard Europarl because of the time requirement for training the proposed PBHSMM. Table 1 gathers some basic statistics of the training partition; and Table 2 is the counterpart for testing.

All the experiments were carried out using a 4-gram language model computed with the standard tool SRILM (Stolcke, 2002), and a modified Kneser-Ney smoothing. To define a translation baseline, we compare our results with

Training	Europarl-10		Europarl-20	
	En	Sp	En	Sp
sentences	$76,996$		$306,897$	
avg. length	7.01	7.0	12.6	12.7
running words	546K	540K	3.86M	3.91M
voc. size	16K	22K	37K	58K

Table 1: Basic statistics of the training sets.

Test	Europarl-10		Europarl-20	
	En	Sp	En	Sp
sentences	$5,000$		$5,000$	
avg. length	7.2	7.0	12.8	13.0
running words	35.8K	35.2K	62.1K	63.0K
ppl (3-gram)	53.4	64.4	77.6	86.8

Table 2: Basic statistics of the test sets.

Iterations	En → Sp		Sp → En	
	TER	BLEU	TER	BLEU
Moses $p(\boldsymbol{x} \mid \boldsymbol{y})$ baseline				
	50.0	32.9	47.2	32.7
Iterations	Baum-Welch			
0	51.4	31.9	48.2	33.2
1	51.4	31.9	47.9	33.1
2	51.5	31.9	47.9	33.1
4	51.2	32.6	48.1	33.1
8	51.4	31.8	48.0	33.0
Iterations	Viterbi			
0	51.4	31.9	48.2	33.2
1	51.4	31.9	47.9	33.1
2	51.1	32.6	48.0	33.2
4	51.2	32.6	48.0	33.0
8	51.4	31.8	48.0	33.0

Table 3: Results obtained with the Europarl-10 corpus with a maximum phrase length of 4.

Moses (Koehn and others, 2007) but constraining the model to only use a phrase-based inverse model.

For evaluating the quality of the translations we have used two error measures: bilingual evaluation understudy (BLEU) (Papineni et al., 2001), and translation edit rate (TER) (Snover and others, 2006).

The proposed training algorithms need an initial guess. To this aim, we have computed the IBM word models alignments with GIZA++ (Och and Ney, 2003), for both translation directions. Then, we have computed the simmetrisation heuristic (Och and Ney, 2004) and extracted all the *consistent* phrases (Och and Ney, 2004). Afterwards, we have computed our initial guess by counting the occurrences of each bilingual phrase and then normalising the counts. Instead of directly using the Moses system to do this work, we have implemented our own version of this process.

Since the training algorithm highly depends on the maximum phrase length, for most of the experimentation we have limited it to 4. In Table 3, the results obtained for both translation directions are summarised for the Europarl-10. Surprisingly, Viterbi training obtains almost the same results that the Baum-Welch training; probably because most of the sentences accumulate all the probability mass in just one possible segmentation. Maybe that is why our algorithm is not able to obtain a large improvement with respect to the initialisation. Note that since the proposed system and Moses use different phrase-tables, the comparison of this two numbers is not fair. Therefore, the Moses baseline is only given as a reference and not as a system to improve. The important question is whether the model produces an improvement with respect to the initialisation, i.e., the result on iteration 0. Note that this corpus is small, and although its complexity allow us to check some PBHSMM properties, we cannot to obtain further conclusions.

On the other hand, Table 4 summarises the results obtained with the Europarl-20. This Table only report results for the Viterbi training since again Baum-Welch training has no advantage with respect to it. Typically, over 4 iterations suffices to avoid over-training, and maximise the system performance. The results show a minor improvement over the initialisation. Although the improvement is small, its magnitude is similar to the improvement obtained when extending the maximum phrase length as shown in Table 5. For instance, it is seen that extending the maximum phrase length from 4 to 5 incurs in the same improvement that performing 4 Viterbi iterations with a maximum phrase length of 4. In most of the cases the Viterbi training improves the translation quality.

Although, in most cases the training does not incur in a significant improvement over the baseline; in practice the quality of the translations is increased by the training. In Table 6, we have selected some translation examples. A detailed analysis of the system translations suggest that most cases belong to the cases A or B.

Case A	Training improves evaluation measures
REF.	I sincerely believe that the aim of the present directive is a step in the right direction .
IT. 0	I am convinced that the aim of this directive is a step in the right direction .
IT. 4	I sincerely believe that the aim of the directive before us is a step in the right direction .
MOSES	I sincerely believe that the aim behind the directive is also a step in the right direction .
Case B	Training improves translation but not evaluation measures
REF.	Mr president , i wish to endorse mr posselt 's comments .
IT. 0	Mr president , i support for to our .
IT. 4	Mr president , i join in good faith to our colleague , mr posselt .
MOSES	mr president , i would like to join in good faith in the words of our colleague , mr rbig .
Case C	Training degrades evaluation measures
REF.	BSE has already cost the uk gbp 1.5 billion in lost exports .
IT. 0	BSE has cost the uk 1.5 million losses exports .
IT. 4	BSE already has cost in the uk alone 1500 million pounds into loss of exports .
MOSES	BSE has already claimed to britain 1500 million pounds into loss of trade .
Case D	Other cases
REF.	Are there any objections to amendment nos 3 and 14 being considered as null and void from now on ?
IT. 0	Are there any objections to give amendments nos 3 and 14 .
IT. 4	Are there any objections to adopt amendments nos 3 and 14 ?
MOSES	Are there any objections to consider amendments nos 3 and 14 ?

Table 6: Some translation examples (Sp → En) before and after training the phrase table 4 iterations with the Viterbi training and maximum phrase length of 4.

Iterations	En → Sp		Sp → En	
	TER	BLEU	TER	BLEU
Moses $p(\boldsymbol{x} \mid \boldsymbol{y})$ baseline				
	57.3	23.5	55.1	24.10
Iterations	Viterbi			
0	57.7	25.0	56.0	26.0
1	57.7	25.1	55.8	26.4
2	57.7	25.1	55.9	26.4
4	57.7	25.2	55.8	26.5
8	57.7	25.2	55.8	26.5

Table 4: Results obtained with the Europarl-20 corpus with a maximum phrase length of 4.

Iterations	En → Sp		Sp → En	
	TER	BLEU	TER	BLEU
Iterations	Maximum phrase length 2			
0	60.5	21.2	57.9	23.5
4	60.5	21.2	58.1	23.5
Iterations	Maximum phrase length 3			
0	58.6	24.1	56.1	25.7
4	58.3	24.1	56.4	25.5
Iterations	Maximum phrase length 4			
0	57.7	25.0	56.0	26.0
4	57.7	25.1	55.8	26.5
Iterations	Maximum phrase length 5			
0	57.7	25.1	55.8	26.6
4	57.4	25.3	55.3	26.9
Iterations	Maximum phrase length 6			
0	57.7	25.4	55.9	26.6
4	57.3	25.6	55.4	26.8

Table 5: Results obtained with the Europarl-20 corpus for several maximum phrase lengths.

5 Conclusions and Future work

We have presented a phrase-based hidden semi-Markov model for machine translation inspired on both phrase-based models and classical hidden semi-Markov models. The idea behind this model is to provide a well-defined monotonic formalism that explicitly introduces the statistical dependencies needed to define the monotonic translation process with theoretical correctness and without moving away from the phrase-based models.

A detailed practical analysis showed a slight improvement by applying the estimation algorithms

with respect to the baseline. Surprisingly, we have observed that both trainings, Viterbi and Baum-Welch, obtain the same practical behaviour. Therefore, we recommend the use of the fastest: the Viterbi training. However, we have not used the proposed PBHSMM as a feature inside a log-linear model as most of the current state-of-the-art systems. We leave this comparison as a future work.

As discussed in section 2, one outstanding and simple extension to the proposed model is to unhide the *concept* variable by having a mixture of phrase-based dictionaries, $\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{y}, c)$. Actually, the requirements of this modification would not significantly affect to the proposed estimation algorithms. We are already extending the model towards this direction.

Finally, the most undesirable property of the proposed model is its monotonicity at phrase level. Although the monotonic constraint is a clear disadvantage for this primer PBHSMM translation model, it can be extended to non-monotonic processes. However, we leave these extensions as future work.

Acknowledgement

Work partially supported by the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018), 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

Alexandra Birch, Chris Callison-Burch, Miles Osborne and Philipp Koehn. 2006. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings of the NAACL 2006 Workshop on Statistical Machine TranslationConference*.

Andrés-Ferrer, J. and A. Juan-Císcar. 2007. A phrase-based hidden markov model approach to machine translation. In *Proceedings of New Approaches to Machine Translation*, pages 57–62, January.

Brown, P. F. et al. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Chen, S. F. and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL'96*, pages 310–318, Morristown, NJ, USA, June. Association for Computational Linguistics.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Ser. B*, 39(1):1–22.

DeNero, J., D. Gillick, J. Zhang, and D. Klein. 2006. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City, June. Association for Computational Linguistics.

Koehn, P. et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL'07: Demo and Poster Sessions*, pages 177–180, Morristown, NJ, USA, June. Association for Computational Linguistics.

Koehn, P., F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL'03*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. of the MT Summit X*, pages 79–86, September.

Marcu, Daniel and Qilliam Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, Philadelphia, PA, July.

Murhpy, Kevin P. 2007. Hidden semi-Markov Models (HSMMs). Technical report, University of British Columbia.

Och, F.J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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

Ostendorf, M., V. Digalakis, and O. A. Kimball. 1996. From hmms to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 4:360–378.

Papineni, K., S. Roukos, T. Ward, and W. Zhu. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176, Thomas J. Watson Research Center.

Rabiner, Lawrence R. 1990. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296.

Snover, M. et al. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA'06*, pages 223–231, Boston, Massachusetts, USA, August. Association for Machine Translation in the Americas.

Stolcke, A. 2002. SRILM – an extensible language modeling toolkit. In *Proc. of ICSLP'02*, pages 901–904, September.

Building Strong Multilingual Aligned Corpora

Reza Bosagh Zadeh
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
rezab@cs.cmu.edu

Abstract

Recent advances have allowed algorithms that learn from aligned natural language texts to exploit aligned sentences in more than two languages. We investigate ways of combining $\binom{N}{2}$ bilingual aligned corpora together to create a multilingual aligned corpus across N languages. As a result of the combination of several corpora, our algorithms output a multilingual corpus, with each aligned tuple assigned a quality score called 'strength' that may be used when learning from the multilingual corpus. We show that the addition of bilingual corpora used with alignment strengths can significantly improve Statistical Machine Translation quality on an Arabic→English task.

1 Introduction

In Machine Translation, it is desirable and intuitive that bridging through languages other than the source and target should help improve translation quality between the source and target. By using bilingual alignments across all pairs of N languages, (Kumar et al., 2007) was able to gain in translation quality between two languages by way of alignment bridges. They describe an approach to improve Statistical Machine Translation (SMT) performance using multi-lingual, parallel, sentence-aligned corpora in several bridge languages. Their approach consists of a simple method for utilizing a bridge language to create a word alignment system and a procedure for combining word alignment systems from multiple bridge languages. They present experiments showing that multilingual, parallel text in Spanish, French, Russian, and Chinese can be utilized to improve translation performance on an Arabic-to-English task. Other papers that use multilingual aligned corpora include (Borin, 2000; Mann and Yarowsky, 2001; Simard, 1999; Kumar et al., 2007). With the advent of such learning strategies requiring a multilingual aligned corpus, it is desirable to take existing bilingual aligned corpora and combine them into a single multilingual aligned corpus.

We investigate ways of combining $\binom{N}{2}$ bilingual aligned corpora together to create a multilingual aligned corpus across N languages. As a result of the combination of several corpora, our algorithms not only output a multilingual corpus, but each multilingual aligned tuple is further assigned a quality score that may be used when learning. We call the alignment quality measure produced by the combination procedure the alignment's **strength**. By using alignment strengths in a setting similar to (Kumar et al., 2007) we show that strengths can be used to improve SMT quality in an Arabic→English task by re-weighing the training corpus to give higher weight to stronger alignments. This allowed us to add more bilingual corpora at training time while consistently improving translation quality.

The problem of creating multilingual alignments from bilingual ones is not trivial. As an example of the problem being solved, consider 4 sentences A, B, C, D in 4 languages. As part of input, alignments are given between each pair of languages. For each alignment between two sentences that exists in the input, an edge is present in Figure 1. How does one deal with the missing alignments (A, D) and (C, D)? Should all 4 sentences be aligned together in the multingual corpus, or

Proceedings of the 13th Annual Conference of the EAMT, pages 176–181,
Barcelona, May 2009

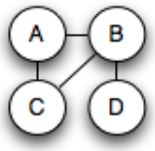


Figure 1: Links missing: (A, D) and (C, D).

should D be separated from the clique created by A, B, C? We explore answers to these questions and seek the best solution for Arabic→English machine translation.

We also present experiments that show using at most one bridge language (i.e. 3 languages in total) provides the optimal quality gain in our experiments when training under the framework presented by (Kumar et al., 2007). Then, using this observation, we can investigate the benefits of alignment strengths.

2 Framework

Each bilingual corpus is defined as a triple $(S_1, S_2, \mathcal{A})$ where S_1 and S_2 are finite sets of all sentences observed in the corpus for language 1 and 2, respectively. $\mathcal{A} \subseteq S_1 \times S_2$ contains alignments where each $(s_1, s_2) \in \mathcal{A}$ aligns sentence s_1 to s_2.

Our algorithms take as input $\binom{N}{2}$ aligned bilingual corpora and output a single multilingual corpus across N languages, where each alignment in the new multilingual corpus is also accompanied by a quality score, its strength. Thus a multilingual aligned tuple is defined as a $(N+1)$-tuple $(S_1, \cdots, S_N, \mathcal{M})$ where

$$\mathcal{M} \subseteq \bigcup_{i=2}^{i=N} S_1 \times \cdots \times S_i \times \mathbb{R}$$

In other words $\mathcal{M}$ is a set of alignments between at most N languages where each alignment is accompanied by a quality score. S_j is the set of all sentences observed for language j. In the case that $N = 2$, a multilingual corpus degenerates to a bilingual corpus.

An important property of our combination algorithm is that all the bilingual tuples from any of the input corpora will still be available to any algorithms that are primarily interested in a bilingual corpus i.e. none of the information available in any of the bilingual corpora is lost. Along with the added advantage of having a quality score and alignments to other languages.

Note that even though we call each collection of alignments a corpus, the same concepts and ideas introduced in this paper can be applied to document-level alignments and even sentence-level alignments, where the documents or sentences are available in more than 2 languages and are to be used for creation of a multilingual document-aligned or sentence-aligned corpus.

3 Combining Corpora

3.1 Problem definition

Given $\binom{N}{2}$ bilingual corpora - all pairs amongst N languages - the goal is to create a single multilingual corpus encapsulating all alignments within the individual corpora. Of the result $(S_1, \cdots, S_N, \mathcal{M})$ it is clear that each of $S_1, \cdots, S_N$ will simply be the union of the S's observed in the bilingual input for each language. The interesting problem is to generate alignments spanning more than two languages.

To answer this question, consider the multipartite graph $\mathcal{G}$ composed of nodes $S_1 \cup \cdots \cup S_N$, where the edge-list is defined as $\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{\binom{N}{2}}$. As a multipartite graph, $\mathcal{G}$ will have N partitions. We wish to produce $\mathcal{M}$ from $\mathcal{G}$ where each $\alpha \in \mathcal{M}$ will have one entry from each partition, along with a quality score. In the case that $N = 2$ it is clear how to generate $\mathcal{M}$, simply take each edge in the graph and add its endpoints as a new element of $\mathcal{M}$; all alignments will have equal strength (defined later), since there is only one set of bilingual alignments.

However, in case $N > 2$ the problem becomes more interesting. Consider $N = 4$. We face the problem of deciding which nodes from each partition should be part of a single 4-tuple. It is clear that every alignment $\alpha \in \mathcal{M}$ should be a connected subgraph of $\mathcal{G}$ on N or fewer nodes with each node in a different partition; it must be connected, otherwise there would be no evidence provided from any input corpora linking two disconnected components. Throughout this paper when we refer to a subgraph of $\mathcal{G}$, we mean a subgraph of $\mathcal{G}$ where all nodes are in different partitions. We also use the concept of a multilingual alignment and subgraphs of $\mathcal{G}$ interchangeably, as there is an obvious bijection between the two in our framework.

It is a subtle but important observation that each

edge from the edgelist must only be involved at most once in a certain alignment α. If this were not the case, then we would be over-representing a particular link between two sentences without justification.

3.2 Combination

Using all and only the information available in $\mathcal{G}$ we wish to produce $\mathcal{M}$. With that goal, we will find and remove the connected subgraphs of $\mathcal{G}$ on n nodes, for some $2 \leq n \leq N$, i.e. remove all edges in the subgraph from $\mathcal{G}$ after adding to $\mathcal{M}$ the n-tuple corresponding to the removed subgraph. Our motivation is to pick the most dense subgraphs first, since the number of edges that are involved in the subgraph counts the number of bilingual aligned corpora from the input that *support* the n-tuple. We say that a bilingual corpus $(S_1, S_2, \mathcal{A})$ *supports* an n-tuple if one of the edges from the n-tuple's subgraph exists in $\mathcal{A}$. If all $\binom{n}{2}$ bilingual corpora support a certain n-tuple, then it is fully connected and a clique on n nodes. Cliques should be removed from $\mathcal{G}$ and considered high-quality alignments since they are supported by all input corpora. However, in our data (Uni, 2006) only 11% of the subgraphs were fully-formed cliques and often the subgraphs are missing edges. We exploit this level of agreement amongst the bilingual corpora to assign each n-tuple a quality score, its strength. The strength of an n-tuple (and consequently an alignment) is tentatively defined (until the next section) as the number of edges involved in the subgraph corresponding to the n-tuple, normalized by the number of possible edges, i.e. the edge density of the graph.

Note that we are only considering *connected* subgraphs on n nodes. To connect n nodes, a minimum of $n-1$ edges are necessary. Thus the strength of any alignment must be at least $\frac{n-1}{\binom{n}{2}} = \frac{2}{n}$, and at most $\binom{n}{2}/\binom{n}{2} = 1$.

In order for the output of our algorithms to entirely encapsulate all alignments from the input corpora, for a given n such that $n < N$, we also have to deal with sentences that are only part of subgraphs with n nodes; we call such tuples *deficient*. To find deficient tuples, we remove all connected N-tuples from $\mathcal{G}$ by order of strength. Then whichever edges remain in $\mathcal{G}$ will be participant in deficient subgraphs. To ensure that such edges are accounted for, we repeat the same removal process of subgraphs; only for $(N-1)$-tuples. Overall, the same procedure is applied in order to $\mathcal{G}$ for N-tuples, $(N-1)$-tuples, $(N-2)$-tuples, $\cdots$, 2-tuples, until all edges are removed. Exhaustion of all edges in $\mathcal{G}$ is guaranteed to happen when 2-tuples are removed.

3.3 Strength defined

With the current definition of strength, it is meaningless to compare the strengths of two tuples that are comprised of different numbers of nodes. Indeed with the current definition of strength our experiments did not yield effective results. For example, 2-tuples always have strength equal to 1, since the only edge that can connect the two nodes is present. We wish to assign higher strengths to alignments that are supported by more input corpora. As an example, a clique on 5 nodes should have higher strength than a 2-tuple, but with the current definition of strength the two tuples will have equal strength. To remedy this situation we redefine strength to take into account the number of nodes that are involved in the n-tuple, normalized by the total number of nodes that could potentially be involved. To achieve such discrimination, we redefine the **strength** $\mathrm{str}(\alpha)$ of an n-tuple α (and consequently an alignment) as the edge density of α, dampened by the fraction of potential languages involved in α. Thus

$$\mathrm{str}(\alpha) = \frac{q}{\binom{n}{2}} \frac{n}{N} = \frac{2q}{(n-1)N}$$

where q is the number of edges in α. With this definition it is now the case that for all α

$$\frac{2}{N}\frac{2}{n} = \frac{4}{nN} \leq \frac{2}{N} \leq \mathrm{str}(\alpha) \leq 1$$

since $\frac{2}{n} \leq \frac{q}{\binom{n}{2}}$ and $2 \leq n \leq N$. Figure 2 shows several alignments with varying strength.

4 Experiments

We now present experiments to demonstrate the advantages of using alignment strengths. We also present experiments that show using at most one bridge language provides optimal quality gain in our experiments. Our experiments are performed in the open data track of the NIST[1] Arabic→English machine translation task.

[1] http://www.nist.gov/speech/tests/mt/

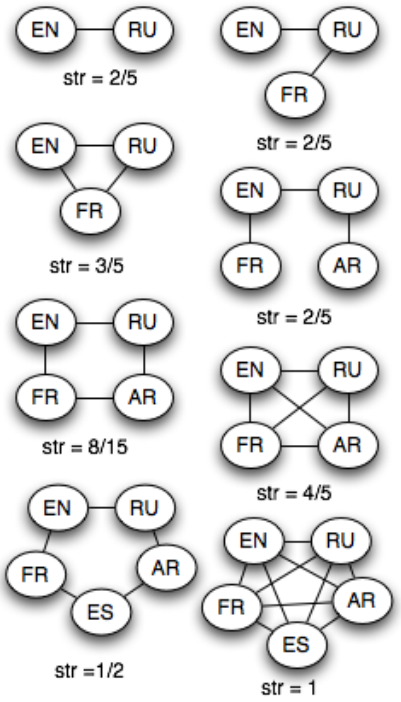


Figure 2: Connected subgraphs of $\mathcal{G}$ and corresponding strengths. Alignment strengths across $N = 5$ languages, $\mathcal{G}$ has 5 partitions. Each node is a sentence in the language labeled.

4.1 Constructing Word Alignment Using a Bridge Language

Once our multilingual corpus was created as described in section 3.2, we have available tuples of sentences that were translations of each other across 3 (or more) languages. The alignment strengths were used to reweigh the corpus, giving higher weight to stronger alignments, achieved by simply multiplying any counts using a tuple by the tuple's alignment strength. We bridged the alignments using the same method of (Kumar et al., 2007). After creating the multilingual corpus, we have triples of sentences that are translations of each other in languages F, E, and the bridge language G: $\mathbf{f} = f_1^J, \mathbf{e} = e_1^I, \mathbf{g} = g_1^K$. We use the notation of (Kumar et al., 2007), where the goal is to obtain posterior probability estimates for the sentence-pair in FE: $(\mathbf{f}, \mathbf{e})$ using the posterior probability estimates for the sentence pairs in FG: $(\mathbf{f}, \mathbf{g})$ and GE: $(\mathbf{g}, \mathbf{e})$. The word alignments between the above sentence-pairs are referred to as $\mathbf{a}^{FE}$, $\mathbf{a}^{FG}$, and $\mathbf{a}^{GE}$ respectively; the notation $\mathbf{a}^{FE}$ indicates that the alignment maps a position in F to a position in E.

Set	# of Ar words (K)	# of sentences
dev1	48.8	2056
dev2	12.5	502
test	39.2	1678
blind	37.1	1799

Table 1: Statistics for the test data.

Under some assumptions, (Kumar et al., 2007) arrive at the final expression for the posterior probability FE in terms of posterior probabilities for GF and EG

$$P(a_j^{FE} = i|\mathbf{e}, \mathbf{f}) = \sum_{k=0}^{K} P(a_j^{FG} = k|\mathbf{g}, \mathbf{f}) P(a_k^{GE} = i|\mathbf{g}, \mathbf{e}) \quad (1)$$

The above expression states that the posterior probability matrix for FE can be obtained using a *simple matrix multiplication* of posterior probability matrices for GE and FG. Similarly, we can obtain posterior probability matrices when more than 3 languages are involved by multiplying several of these matrices together.

Next we need to combine word alignment posterior probability matrices from many different bridges, along with the direct alignments posterior matrix. Suppose we have translations in bridge languages $G_1, G_2, ..., G_N$, then we can generate a posterior probability matrix for FE using one or more of the bridge languages. In addition, we can always generate a posterior probability matrix for FE with the FE alignment model directly without using any bridge language. These posterior matrices can be combined by simple interpolation. Instead of simple interpolation, one could also combine the matrices with specific weights given to path, but we leave that for future work.

4.2 Training and Test Data

We train alignment models using the Official Document System of the United Nations parallel data (Uni, 2006). This data-set contains documents from the parliament from 1993 onwards. The corpus is parallel across the six official languages of the United nations: Arabic (AR), Chinese (ZH), English (EN), French (FR), Russian (RU), and Spanish (ES).

To create test sets, we follow the same strategy as (Kumar et al., 2007) and combine the NIST 2001-2005 Arabic-English evaluation sets into a pool, that is randomly sampled into two development sets (dev1, dev2) and a test set (test) with

2056, 502, and 1678 sentences respectively. Our blind test (blind) set is the NIST part of the NIST 06 evaluation set consisting of 1799 sentences. We report results on the blind set. Some statistics computed on the test data are shown in Table 1. Significance was tested using a paired bootstrap (Koehn, 2004) with 1000 samples ($p < 0.05$). BLEU scores in bold are significantly different from the baseline.

4.3 Phrase-based SMT system

We use a phrase-based SMT system following the ideas of (Och and Ney, 2004). First, a list of phrase-pairs up to length 7 is obtained from word alignments. Features (Och and Ney, 2004) are computed over the phrase table. An n-gram word language model for English is trained on a monolingual corpus. Finally, Minimum Error Rate Training (Och, 2003) for the BLEU (Papineni et al., 2002) quality metric is used to estimate 20 feature weights over dev1. For decoding we use a standard dynamic programming beam-search decoder (Och and Ney, 2004). A two stage process is used; first an inventory of the 1000-best hypotheses is produced, which is then reordered using Minimum Bayes-Risk (MBR) decoding (Kumar and Byrne, 2004). The MBR scaling parameter is tuned on dev2.

4.3.1 Results

By multiplying several bridged posterior probability matrices, we can create bridges of lengths greater than 3. For example, for translating from F to E using two bridge languages G_1, G_2 we can produce alignment posterior matrices for FG_1, G_1G_2, G_2E and use these to produce $FE = FG_1 \times G_1G_2 \times G_2E$. In table 2 we see that using at most 1 bridge language is the best bridging strategy. This is because the noise introduced by bridging through a second language outweighs any benefits gained by bridging through a single language. Note that in table 2 each row numbered n corresponds to using all bridges of length n, of which there are exponentially many. However, this is not a problem as $n \leq 4$.

Given the results in table 2, we restrict our next experiments to bridging through a single language, as that provides the best gain in our experiments. Using alignment strengths helps us to consistently add more bilingual corpora while maintaining or increasing quality. Table 3 shows the gains obtained by adding bilingual corpora involving languages other than AR and EN.

# of bridges	AR-EN BLEU (%)
0	38.2
1	**39.2**
2	38.1
3	**37.9**
4	**37.8**

Table 2: Results on the blind set. Each row n corresponds to combining all bridges of length n. Using exactly all bridges of length 1 is optimal for our experiments.

# of languages	AR-EN BLEU (%)
2 (AR, EN)	38.2
3 (+ ES)	**38.7**
4 (+ FR)	**38.9**
5 (+ RU)	**39.1**
6 (+ ZH)	**39.1**

Table 3: Results on the blind set. Each row adds new bilingual corpora to the corpora from the previous row.

5 Conclusions and Future work

We have presented a method to combine bilingual aligned corpora into a multilingual aligned corpus in a nontrivial way. We defined the strength of a multilingual alignment as a metric proportional to the edge density of the alignment. By using alignment strengths, we observed gains in Arabic→English machine translation quality. By adding further bilingual corpora, we show that alignment strengths can be used to consistently better translation quality. We also noticed that using at most one bridge language is optimal in our experiments.

While all of our work is focused on machine translation, the simple of idea of reweighing the training corpus according to alignment strengths can be applied to other problems where a multilingual corpus is useful. Also, there is potential for alignment strengths to be used at other points in the training pipeline, e.g. during word alignment.

Acknowledgments

The author was supported by a fellowship from the National Sciences and Engineering Research Council of Canada and project funding on a National Science Foundation grant.

References

Borin, L. 2000. You'll take the high road and I'll take the low road: using a third language to improve bilingual word alignment. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 97–103. Association for Computational Linguistics Morristown, NJ, USA.

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In Lin, Dekang and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Kumar, Shankar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In Susan Dumais, Daniel Marcu and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 169–176, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Kumar, Shankar, Franz J. Och, and Wolfgang Macherey. 2007. Improving word alignment with bridge languages. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 42–50, Prague, Czech Republic, June. Association for Computational Linguistics.

Mann, Gideon S. and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

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

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Simard, M. 1999. Text-translation alignment: Three languages are better than two. In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 2–11.

United Nations, 2006. *ODS United Nations Parallel Corpus*. http://ods.un.org/.

A Constraint Satisfaction Approach to Machine Translation

Sander Canisius[*,**] and **Antal van den Bosch**[*]
[*]Tilburg centre for Creative Computing, P.O. Box 90153, 5000 LE Tilburg, The Netherlands
[**]The Netherlands Cancer Institute, P.O. Box 90203, 1006 BE Amsterdam, The Netherlands
S.Canisius@nki.nl, Antal.vdnBosch@uvt.nl

Abstract

Constraint satisfaction inference is presented as a generic, theory-neutral inference engine for machine translation. The approach enables the integration of many different solutions to aspects of the output space, including classification-based translation models that take source-side context into account, as well as stochastic components such as target language models. The approach is contrasted with a word-based SMT system using the same decoding algorithm, but optimising a different objective function. The incorporation of source-side context models in our model filters out many irrelevant candidate translations, leading to superior translation scores.

1 Introduction

The vast complexity of the translation task has led designers of machine translation systems to delegate the task to multiple submodels. The crucial task in these systems is the integration of all available information in the best possible way. In this paper we advocate and present an eclectic, theory-neutral approach to machine translation that employs *constraint satisfaction* as the integrator method, and that regards the translation task as a *structured prediction problem*.

Structured prediction is a relatively new and emerging field in machine learning in which generic techniques are developed that explicitly model structural properties of the output space (Bakir et al., 2007). Statistical machine translation systems can be seen as a forebearer of this field; conditional random fields (Lafferty et al., 2001) and Searn (Daumé III, 2006) are more recent exponents of the approach. The typical solution to a structured prediction problem is to regard it as a combinatorial optimisation in an output space that spans all possible outputs for a given input. One or more learning components are responsible for learning (parts of) an objective function, and a search (or inference) component finds the output structure that maximises the objective function.

Within statistical machine translation systems, the probabilistic underpinning of all involved components acts as a kind of industrial standard. This has the positive effect that a substantial body of work could be built using the same universal language. At the same time, it makes the integration of non-stochastic components into statistical machine translation systems sometimes unwieldy. Yet, recent experiments on mixing local classifications of non-stochastic machine learners with statistical MT models (Carpuat and Wu, 2007; Stroppa et al., 2007), has shown its potential. Arguably, it makes sense to investigate the gathering of a mix of views on the objective function as input to the final search or inference process, and this mix should be eclectic—that is, theory-neutral—to allow extremely different but successful partial solutions to the objective function to participate.

As we argue in Section 2, the classic constraint satisfaction framework offers the right apparatus to offer such a theory-neutral basis. In Section 3 we describe our experimental setup; the outcomes of a comparative study with an unconstrained but otherwise equivalent word-level SMT system are discussed in Section 4. In Section 5 we formulate our conclusions.

2 Constraint satisfaction

In constraint satisfaction (Tsang, 1993) the goal is to find values for a set of variables that satisfy certain constraints. While a variable's domain dictates the values a single variable is allowed to take, the constraints of a constraint satisfaction problem specify which *simultaneous value combinations* over a number of variables are allowed. Here, we

Proceedings of the 13th Annual Conference of the EAMT, pages 182–189,
Barcelona, May 2009

adopt a weighted constraint satisfaction approach. Candidate solutions to a weighted constraint satisfaction problem are scored according to the sum of weights of the constraints they satisfy, and the highest scoring solution is selected.

The constraint satisfaction approach presented here is formulated as an extension of statistical log-linear models (Berger et al., 1996; Papineni et al., 1998)[1]. A typical log-linear model for machine translation combines a number of feature functions, each of which measures the quality of a candidate translation according to some aspect. Two feature functions tend to be part of any SMT system; a translation model (TM), and a target language model (LM), measuring the faithfulness and the fluency of the translation, respectively. Both are probability distributions, obtained by maximum-likelihood estimation from training data. The best translation is determined by maximising a weighted sum of those feature functions:

$$\operatorname*{argmax}_{y} \lambda_{\text{TM}} \log P(x|y) + \lambda_{\text{LM}} \log P(y)$$

One of the problems with this traditional formulation is that the translation model ignores the context in which it is applied. This is the case for the source sentence context, as well as for the target sentence context. It is expected that the language model compensates for this. However, it is questionable whether this is a reasonable expectation. Since the language model does not take into account the source side at all, it can only resolve source-side ambiguities indirectly by looking at the translations of source words. This means that the language model is not only used for attaining good fluency, but also in part for attaining good faithfulness, the latter of which it might not be good enough for. Our extension uses constraint satisfaction to improve the translation model, by having it take into account both source and target sentence contexts.

To do this, we replace the language model by a *constraint model*. The score assigned to a candidate translation by this model corresponds to the sum of weights of satisfied constraints according to a constraint satisfaction problem. The score formula is adapted as follows:

$$\operatorname*{argmax}_{\mathbf{y}} \lambda_{\text{CM}} f_{\text{CM}}(\mathbf{y}) \quad \textbf{(1)}$$

$$+ \lambda_{\text{LM}} \log P(\mathbf{y}) \quad \textbf{(2)}$$

$$+ \lambda_{\text{NM}} \sum_{i} [y_i = \emptyset] \log P(y_i = \emptyset | x_i) \quad \textbf{(3)}$$

$$+ \lambda_{\text{LP}} |\mathbf{y}| \quad \textbf{(4)}$$

The constraint model feature function (1, CM) scores the satisfied soft constraints. Considering the difficulty of the translation task, we augment the objective function with three more feature functions. The language model (2, LM) is a standard back-off trigram language model, estimated using the SRILM toolkit (Stolcke, 2002). Two more feature functions are intended to compensate for the effect that n-gram language models tend to prefer shorter translations. The first, which we call the null model (3, NM), multiplies the translation probability of those source-language words that are translated to $\emptyset$, i.e. words for which no corresponding word is generated in the target-language sentence. It is estimated using relative frequencies, and is in fact similar to the translation model of SMT systems, with the exception that it only applies to source-language words left untranslated. The final feature function, the length penalty (4, LP), counts the number of target-language words. Given a positive weight λ_{LP}, it is in fact a length bonus rather than a penalty.

For the implementation of the constraint model, we define a weighted constraint satisfaction problem over a solution space of possible translations. Therefore, two questions need to be answered. First, how do we restrict the solution space? We aim at excluding most candidate solutions before the inference even starts. Defining this solution space is done by introducing variables and populating their domains, and by formulating certain hard constraints that every valid translation has to satisfy. Second, what soft constraints are added to the constraint satisfaction problem? We would like those constraints to improve the faithfulness of the translation by taking into account both source sentence context and target sentence context.

2.1 Solution space

Efficient approaches to machine translation have to make strong assumptions about the parts of the output space that are actually worth exploring. The approach presented here is sufficiently restricted

[1]The log-linear formulation of the objective function can be shown to be equivalent to a weighted constraint satisfaction problem, but we choose to follow this formulation, since it eases the comparison with SMT systems.

to allow for efficient decoding, while remaining expressive enough to attain good translation performance. To represent a solution space, we start by distinguishing two sub-problems that have to be solved as part of the translation task. First, source-language words have to be translated to the correct target-language word. Secondly, the translated words may need to be reordered—possibly new words have to be inserted as well—to make the translation a natural sentence according to the target language.

2.1.1 Representing word translations

For modelling the translation of words in the source sentence, we adopt the assumption that each word in the source sentence is translated to exactly one word in the target sentence. In our constraint satisfaction framework it is naturally represented by introducing one variable for each source word. During inference, the target-language words that are part of the domain of a variable will be considered as possible translations of the corresponding source word. If domains are constructed simply by listing all possible translations for the given source word as found in the training corpus, the solution space of our model would be rather similar to that of word-based SMT systems. In constraint satisfaction inference, however, we employ classifiers to predict the translations to consider. These predictions implicitly filter out all other possible outcomes, rendering the solution space potentially much smaller. We elaborate on how this is done later in the paper.

A few additional issues need to be dealt with. First of all, spurious words in the source sentence should not be translated to a target-language word. This is resolved by translating the word to a special $\emptyset$ symbol instead. By definition, this $\emptyset$ symbol will be part of the domain of all variables. As a result, any source-language word may be left untranslated in the target translation. A second issue is the fact that several source-language words might be translated by only one target word. In this case, all corresponding variables are assigned that same word. The fact that those matching words are actually a single token in the target sentence is dealt with in the target sentence realisation.

2.1.2 Representing target sentence realisation

Target sentence realisation involves three differences between the source language and the target language that have to be dealt with and represented: (1) word order differences; (2) zero-fertility words, and (3) multi-fertility words.

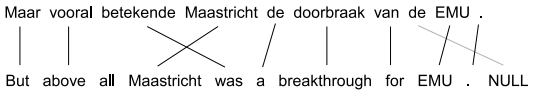


Figure 1: Example of a Dutch-English aligned sentence pair.

Word order differences To cope with arbitrary word reorderings in the translation, the inference procedure needs to consider every permutation of translated source words. For a compact representation of the search space yielded as a result of this, consider a complete directed graph in which the words of the source sentence are represented by vertices, and one additional vertex v_0 corresponds to the start and end of the sentence. A directed arc from vertex v_i to vertex v_j means the translation of the word corresponding to v_j directly follows the translation of the word represented by v_i. In addition, a directed arc from v_0 to v_i, or from v_i to v_0 means that the translation of the word corresponding to v_i is the first or last word of the sentence respectively. The space of all candidate translations corresponds to all cycles that start and end at v_0. Such a cycle is not required to visit every vertex in the graph, i.e. it does not have to be a hamiltonian cycle. Cycles that do not visit a certain vertex v_i correspond to translation candidates in which the source word represented by v_i is not translated. In that case, the translation variable corresponding to this source word should have the value $\emptyset$, which can easily be enforced by a hard constraint.

Given this graph representation of the candidate translation space, it can be reformulated for the constraint satisfaction framework by introducing a set of $(n+1) \times (n+1)$ variables, where n is the length of the source-language sentence, that correspond to the adjacency matrix of the graph just introduced. The domains of all those variables comprise two values, signalling whether or not the corresponding arc is included in the candidate translation cycle. Appropriate constraints have to be added to the constraint satisfaction problem to ensure that every candidate considered is indeed a cycle of the graph. Informally, this is the case if for every $i \in \{1, 2, \ldots, n\}$, either the ith row and column do not contain any positive value at all, or they both contain exactly one positive value. Moreover, the 0th row and column *should* contain exactly one

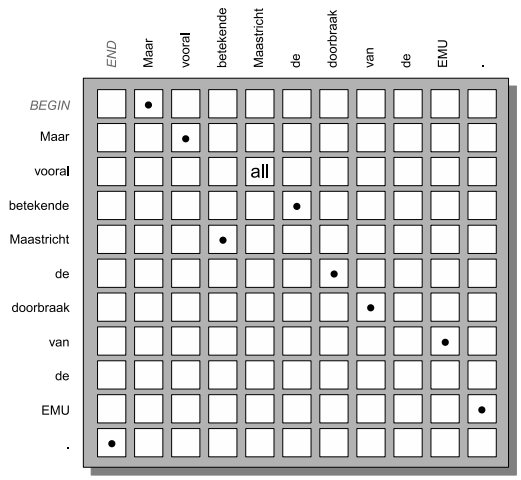


Figure 2: Visualisation of the connectivity matrix of the path corresponding to the correct translation of the Dutch sentence in Figure 1. The "•" in the row labelled with "Maar" and the column labelled with "vooral" denotes that the translation of the latter follows that of the former in the English target sentence. The word "all" is a zero-fertility word.

positive value. To illustrate all of the above, Figure 2 shows a matrix representing the correct translation order for the sample sentence in Figure 1.

Zero-fertility words A common approach to zero-fertility word insertion is to keep a list of frequent zero-fertility words and attempting to insert words from this list at arbitrary positions in the translation. This leads to a substantial expansion of the output space, which was already large to start with. As an alternative to common practice, we choose to attempt insertion of zero-fertility words only if there is evidence that doing so would make sense in the context of the current sentence. This evidence is to be provided by classifier predictions. More specifically, those predictions will be used to collect zero-fertility words that are candidates for insertion between the translations of two source words. In our representation at most one zero-fertility word can separate two fertile words.

Modelling this is possible by a straightforward extension of the adjacency matrix introduced before. In addition to the two values that signify whether or not two translated words are adjacent in the target sentence, a matrix element can also be assigned a word. Such an assignment encodes the case where two translated words are separated by the zero-fertility word stored in the matrix. In Figure 2, the matrix element that connects the translations of "vooral" and "Maastricht" has the value "all", which denotes that the translation of the latter follows that of the former, separated by the word "all". The constraints that ensure that only cycles of the order graph are considered as candidate solutions can be extended easily to this new setting. The zero-fertility word values can simply be treated as positive values.

Maar		But	above
vooral	But	above	all
betekende	Maastricht	was	a
Maastricht	all	Maastricht	was
de	was	a	breakthrough
doorbraak	a	breakthrough	for
van	breakthrough	for	EMU
de			
EMU	for	EMU	

Figure 3: The Dutch example sentence from Figure 1 and the English trigrams that are to be predicted for the words in the sentence. No training example is created for the word "de", because it is aligned with the ∅ token.

Multi-fertility words To account for many-to-one mappings, i.e. mappings of more than one source word to a single target word, we introduce one final value that can be assigned to order matrix entries, signalling overlap between two source words mapped to the same target word.

2.2 Constraints

We choose the constraints for machine translation to cover up to three consecutive target-language words. These constraints are created by predicting a trigram of target-language words for each word in the source sentence. Figure 3 illustrates this process for the sentence pair in Figure 1. The middle word of the predicted trigrams is the hypothesised translation of the source word in focus. The left and right parts are the words surrounding the translation in the target sentence. Note that no training example is created for the Dutch word "de" in Figure 1. Nevertheless, when translating a sentence, trigrams are predicted for all words in the source-language sentence—whether a word is aligned with ∅ is unknown for new sentences.

Given a predicted trigram, two types of constraints are extracted from it. First, a trigram constraint covering the translated word and the two

words surrounding it in the target-language sentence. Secondly, two bigram constraints defined on the translated word and either one of the two surrounding words.

2.3 Solving the CSP

The solution space of the constraint satisfaction problem defined in this section has immense proportions. Even if a base classifier perfectly predicts the correct translations of all source words, which is already overly optimistic, the inference procedure still has to consider every possible permutation of those translated words as a candidate translation. Unfortunately, no further restrictions or assumptions can be made that would restrict the solution space sufficiently to allow for exhaustive solving. Approximate solving is the only option.

With this in mind, we choose the greedy decoding algorithm of Germann (2003) as the basis for the constraint solver. The algorithm starts with a complete candidate translation; for example, one where all source words are mapped to their most likely translations and added to the target sentence in original order. Subsequently, a hill-climbing search is started in which simple transformations of the current translation are attempted and the one leading to the highest score increase is actually applied. New transformations are tried until no further improvement can be attained. The following transformations are considered:

- *Change* the translation of a source-language word. If the target word currently aligned with it has a fertility greater than one, a new target word is inserted in the translation at the position maximising the translation score; otherwise, the current translation is changed, while its position is left unchanged. Among the translation candidates tried is also $\emptyset$, which results in the word being removed from the candidate translation.
- *Insert* a zero-fertility word. According to our model, zero-fertility words are only inserted in between two fertile words.
- *Erase* a zero-fertility word.
- *Join* two target-language words, i.e., removing one of the words from the translation and aligning with the remaining word all words previously aligned with the word that was removed.
- *Swap* two non-overlapping segments of the target sentence.

Although the algorithm has been proposed in the context of statistical machine translation, it can more generally be seen as optimising an arbitrary objective function defined over candidate translations. By replacing the noisy-channel equations optimised originally by a credit function based on constraint weights, the algorithm can be employed for solving our constraint satisfaction problem.

3 Experimental setup

3.1 Data

For our study we use four different corpora covering a diverse range of genres. From each of the four corpora, we prepare data sets for the translation pair Dutch to English:

EuroParl The EuroParl corpus (Koehn, 2005) is a multi-lingual parallel corpus extracted from the proceedings of the European Parliament. The Dutch-English parallel subcorpus consists of 1,313,111 sentence pairs.

JRC-Acquis The JRC-Acquis corpus (Steinberger et al., 2006) comprises a large collection of legislative texts extracted from the Acquis Communautaire. The Dutch-English parallel subcorpus provides 1,235,878 bilingual sentence pairs.

EMEA The EMEA data set is composed of texts made available by the European Medicines Agency. It is one of the corpora included in the OPUS parallel corpus (Tiedemann and Nygaard, 2004). The parallel texts for Dutch and English cover 751,602 sentence pairs.

OpenSubtitles The OpenSubtitles corpus, also part of OPUS, provides aligned movie subtitles in various different languages. For the language pair Dutch-English, it comprises 288,160 sentence pairs.

In four experiments, the translation system has been trained and tested on texts within the same corpus. For this evaluation, as well as for tuning the system, from each of the four corpora, two sets of 1,000 sentences each have been selected for testing and development purposes respectively; the remainder is used for training. This training data has subsequently been aligned at the word level using GIZA++ (Och and Ney, 2000).

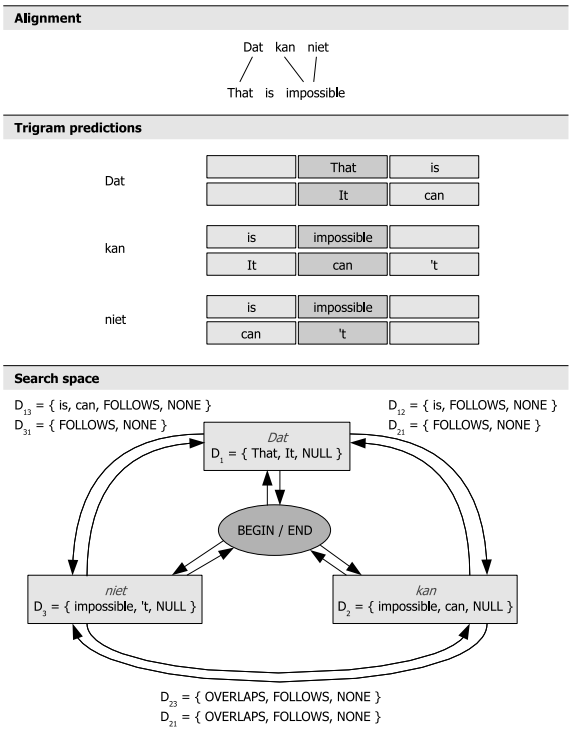


Figure 4: Visualisation of the search space resulting from a set of base classifier predictions. Top: the correct alignment of a Dutch-English example sentence pair. Middle: trigram predictions for the words in the Dutch sentence. Bottom: the complete graph connecting all Dutch words. Any valid translation is a directed cycle in this graph that starts and ends in the BEGIN/END node.

3.2 Constraint prediction

For predicting the soft constraints of our translation model, we need to map each word of the source sentence to a trigram of target words. The middle word of that trigram is the translation of the source word in focus; the left and right words are the two target words surrounding it.

Several recent studies (Carpuat and Wu, 2007; Chan et al., 2007; Giménez and Màrquez, 2007; Stroppa et al., 2007) have experimented with classification-based alternatives to traditional translation models that take into account contextual information of the word in the source sentence, similarly to the way word-sense disambiguation is performed. Our constraint predictor is similar to the classifiers used in these studies in the sense that contextual information is used to improve the suggested translations.

We follow (Stroppa et al., 2007) in using the k-nearest neighbor classifier as implemented in the TiMBL software package (Daelemans et al., 2007). The feature set used in our classifier is simpler, though. In specific, the features used correspond to a word window of length three centred on the focus word. As a consequence of the small number of features and the large number of classes, it will often be the case that the classifier finds several classes that have the same score for an input sentence. Classes that are assigned the same score by a base classifier are the perfect example of uncertainty that cannot be resolved locally, and thus should be delegated to the inference procedure. Therefore, for the experiments described in this paper, we disable tie-breaking in the base classifier, and extract domain values and constraints from all classes that have the maximum score.

The target-word trigrams predicted by the base classifier are used to add constraints to the inference, as well as to compose the domains of the variables. Constraints are derived from the predicted trigrams: the predicted trigram itself is turned into a constraint, but also the two bigrams covered by the predicted trigram.

The constraint satisfaction inference procedure is illustrated in Figure 4, where in the absence of tie-breaking in the classifier, two trigrams have been predicted for each source-language word. Since for all words, both trigrams suggest a unique translation, the domains of the three words, $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$, each contain two candidate translations, as well as the symbol $\emptyset$, which is always included as a possible translation.

The variable domains for the order variables always contain at least the two values that signal that the two corresponding words do or do not follow one another in the translated sentence. In Figure 4, these variables correspond to the edges of the graph depicted at the bottom of the figure. The two symbols `FOLLOWS` and `NONE` are included in all domains. Furthermore, the model also allows for an overlap value or a zero-fertility word as value for order variables. As for the former, the overlap value is only added to the domain of the order variable y_{ij} if words i and j can be translated to the same target word, or more formally, if their domains overlap, $\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset$. As an illustration, $\mathcal{D}_2$ and $\mathcal{D}_3$ both contain the word "impossible", and therefore, $\mathcal{D}_{23}$ and $\mathcal{D}_{32}$ contain the symbol `OVERLAPS`.

Potential zero-fertility words are added to a domain only if base classifier predictions provide sufficient evidence for that. Specifically, the zero-

fertility words included in the domain of the order variable y_{ij} are those words that appear both as the right part of the trigram predicted for word i, and as the left part of the trigram for word j. In the example, the words "That is" predicted for "Dat" overlap with the words "is impossible", predicted for both "kan" and "niet". For this reason, "is" is made a potential zero-fertility word if the translation of either "kan" or "niet" were to follow that of "Dat". Similarly, "can" is a potential zero-fertility word between the translations of "Dat" and "niet", since "It can" has been predicted for the former, and "can 't" for the latter source-language word.

4 Results

To evaluate our constraint satisfaction approach to machine translation, we trained and tested the system using the four Dutch to English data sets described in Section 3.1. In addition, we implemented a word-based SMT system based on the ISI ReWrite decoder, which uses the greedy decoding algorithm of Germann (2003). Comparing with this system is especially interesting since the decoding algorithm is the same as the one used in our constraint satisfaction system. Therefore, the differences that are observed can be attributed to the modelling choices underlying the two systems. First, the constraint satisfaction system uses a richer objective function based on the constraint model that replaces the translation model. Second, constraint satisfaction inference searches a smaller solution space than the ReWrite system, which does not restrict its solution space in advance.

Table 1 lists the BLEU scores (Papineni et al., 2002) and exact Meteor scores (Banerjee and Lavie, 2005) for both systems on each of the four data sets. The two systems are closest in performance on the EuroParl data, though constraint satisfaction inference outperforms ReWrite in terms of both metrics. On EMEA, ReWrite outperforms constraint satisfaction inference; on JRC-Acquis and OpenSubtitles, constraint satisfaction inference outperforms ReWrite again.

The relative performance differences are rather diverse among the four data sets. This may be attributed to the underlying search algorithm, a greedy hill-climbing search, which is known to risk ending up in suboptimal local optima. Constraint satisfaction inference seems to deal with this circumstance better than ReWrite. On the one hand, the richer objective function used by constraint satisfaction inference, based on the predicted constraint model, may account for the better performance of constraint satisfaction inference. On the other hand, though, the smaller solution space searched by constraint satisfaction inference may also be expected to have fewer local optima.

The fact that on the EMEA corpus the ReWrite system performs better may be rooted in the fact that sentences in EMEA are largely formulaic and on average rather short: 9 tokens. Apparently, the hill-climbing algorithm only needs a few transformation operations to reach good translations. Constraint satisfaction inference's objective function causes it to perform more transformation operations than it should.

5 Conclusions

Machine translation systems deal with huge output spaces that are costly to search. Their translation quality depends strongly on the quality of the inference imposed on the search in the output space. One strategy, presented in this paper, is to feed a theory-neutral inference mechanism, constraint satisfaction inference, with several different inputs of arbitrary types.

The decoding algorithm chosen for the experiments in this paper is an important ingredient for achieving the above objective. Since the algorithm is a local hill-climbing method, at any moment at which the objective function evaluates a hypothesis, there is a complete, rather than a partial translation as would be the case in A* or Viterbi search. As a result, the objective function can take into account arbitrary structural dependencies. The possibilities for such dependencies are virtually unlimited. In this paper, we experimented with only one type of constraint, which models trigrams of target-language words. We expect that large improvements can be achieved by introducing additional constraints. For example, constraints that model phrase-based translations, word reordering in the target sentence, or explicit syntactic structure of the target sentence.

A potential weakness caused by using a greedy search method is the risk of ending up with suboptimal solutions as a result of local optima in the search space. Although there is nothing that can really be done about this, one can make sure that the search space in which the decoder operates already has a certain minimum quality. Our constraint satisfaction inference approach uses a

	EuroParl		JRC-Acquis		EMEA		OpenSubtitles	
	BLEU	Meteor	BLEU	Meteor	BLEU	Meteor	BLEU	Meteor
ReWrite	0.198	0.449	0.450	0.611	**0.395**	**0.650**	0.083	0.304
CSI	**0.211**	**0.469**	**0.513**	**0.650**	0.302	0.540	**0.200**	**0.444**

Table 1: BLEU and Meteor (exact) scores for constraint satisfaction inference and the ReWrite SMT system on the four Dutch to English translation tasks.

context-model classifier to define the exact solution space searched by the decoder. As the most important benefit of this, all candidate translations that are part of the solution space are predicted and filtered based on the context of the source-language word in the input sentence. The intended effect is that candidate translations that are irrelevant for the current sentence are not considered by the decoder, and thus local optima based on such translations are made impossible. Results from our comparative experiment show that this effect can indeed be attained.

Acknowledgements

This study was funded by the Netherlands Organisation for Scientific Research, as part of NWO IMIX and the Vici *Implicit Linguistics* project.

References

Bakir, G., T. Hofmann, B. Scholkopf, A. Smola, B. Taskar, and S. Vishwanathan. 2007. *Predicting Structured Data*. The MIT Press, Cambridge, MA.

Banerjee, S. and A. Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Berger, A., S. Della Pietra, and V. Della Pietra. 1996. Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1).

Carpuat, M. and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72.

Chan, Yee Seng, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 33–40.

Daelemans, W., J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 2007. TiMBL: Tilburg Memory Based Learner, version 6.1, reference guide. Technical Report ILK 07-07, ILK Research Group, Tilburg University.

Daumé III, H. 2006. *Practical Structured Learning Techniques for Natural Language Processing*. Ph.D. thesis, University of Southern California.

Germann, U. 2003. Greedy decoding for statistical machine translation in almost linear time. In *Proceedings of the 2003 Human Language Technology Conference, NAACL-HLT 2003*, pages 1–8.

Giménez, J. and L. Màrquez. 2007. Context-aware discriminative phrase selection for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 159–166.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit X*, pages 79–86, Phuket, Thailand.

Lafferty, J., A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA.

Och, F.J. and H. Ney. 2000. Giza++: Training of statistical translation models. Technical report, RWTH Aachen, University of Technology.

Papineni, K, S. Roukos, and R Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proc. Intern. Conf. on Accoustics, Speech, and Signal Processing*, pages 189–192.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Steinberger, R., B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, and D. Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, pages 2142–2147.

Stolcke, A. 2002. SRILM: An Extensible Language Modeling Toolkit. In *Seventh International Conference on Spoken Language Processing*, pages 901–904.

Stroppa, N., A. Van den Bosch, and A. Way. 2007. Exploiting source similarity for SMT using context-informed features. In *Proceedings of the 11th International Conference on Theoretical Issues in Machine Translation (TMI 2007)*, pages 231–240.

Tiedemann, J. and L. Nygaard. 2004. The OPUS corpus-parallel & free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pages 26–28.

Tsang, E. 1993. *Foundations of constraint satisfaction*. Academic Press, San Diego, CA.

Introducing the Autshumato Integrated Translation Environment

Hendrik J. Groenewald
Centre for Text Technology (CTexT)
North-West University
Potchefstroom, South Africa
handre.groenewald@nwu.ac.za

Wildrich Fourie
Centre for Text Technology (CTexT)
North-West University
Potchefstroom, South Africa
wildrich.fourie@nwu.ac.za

Abstract

Translation is an indispensable process for socioeconomic and cultural development in a multilingual society. The use of translation tools such as translation memories and machine translation systems can be beneficial in supporting the human translator. Unfortunately, the availability of these tools is limited for resource-scarce languages. In this paper, we introduce the *Autshumato Integrated Translation Environment* that facilitates a comprehensive set of translation tools, including machine translation and translation memory. The *Autshumato Integrated Translation Environment* is specifically developed for translation between the official South African languages, but it is in essence language-independent and can therefore be used to translate between any language pair.

1 Introduction

South Africa is a culturally diverse country, with a large number of different ethnic groups. This rich cultural diversity is the main reason for South Africa having no fewer than eleven official languages. The South African Constitution (Republic of South Africa, 1996) guarantees equal status to each of the eleven official languages.

Translating between these eleven languages is an immense task, and unsurprisingly, English often acts as the primary language for government communication. The problem with English being the lingua franca is that the other 10 official languages are marginalised. An example of this is the fact that the South African government does not have the capacity to produce parliamentary records in all eleven official languages. Multilingual parliamentary records are one of the most important resources for machine translation (MT) projects elsewhere in the world. One such a project is the *EuroMatrix* project, a statistical and hybrid machine translation project for translating between European languages (see www.euromatrix.net). The *EuroMatrix* project uses the parliamentary records of the European Parliament, available in all 23 official languages of the European Union, as one of their main resources for the development of machine translation systems. The parliamentary records of the South African government are only recorded in English on both provincial and national level. The result of this is that a large number of South African citizens are denied access to important government information in their home language, since English is the mother tongue of only 8.2% of the South African population (Van der Merwe and Van der Merwe, 2006). Another negative aspect is that researchers and developers working on the development of machine translation systems for South African languages do not have access to as large amounts of data as their European colleagues.

The South African Government is aware of the importance of elevating the status and promoting the equal use of all eleven official languages on all levels. As a result of this, the government is involved in a large number of initiatives and projects to promote multilingualism. One such as project is the *Autshumato*[1] Project that was

[1] Autshumato was a Khoi-khoi leader that worked as an interpreter between the Europeans and the Khoi-khoi people during the establishment of the Dutch settlement at the Cape of Good Hope in the 17th century (Giliomee and Mbenga, 2007). Autshumato can for this reason be viewed as one of the first translators in South Africa.

Proceedings of the 13th Annual Conference of the EAMT, pages 190–196,
Barcelona, May 2009

commissioned by the South African Department of Arts and Culture for the development of translation tools and resources for the eleven official South African languages. An integral part of the *Autshumato* Project is the *Autshumato Integrated Translation Environment* (ITE), which is the subject of this paper.

The rest of this article is organised as follows: the next section provides information about the end-user requirements for the *Autshumato* ITE; Section 3 describes related work; and Section 4 provides detailed information about the design and implementation of the system. The main functionalities are discussed in Section 5, while the paper concludes with a discussion of future work in Section 6.

2 End-user requirements

As indicated in the previous section, the South African government's translation agencies do not have the capacity to translate all government documents and communications into all of the official languages. The magnitude of this problem is increased by the lack of availability of translation tools for the eleven official languages. The purpose of the *Autshumato* project is to develop tools and resources that will help government translators increase both the quantity and the quality of their translation work. Although these tools are specifically developed for use by government translators, it will in future be released under an open source license for use by all translators.

Since government translators are the most important clients of the *Autshumato* Project, one of the first objectives of this project was to determine the end-user requirements of the translators working at various government departments. The end-user requirements were elicited by means of questionnaires and joint-requirements planning (JRP) sessions. The joint requirements planning sessions were held with translators working at the offices of the South African National Language Service (NLS) and members of the development team. The results of the questionnaires and joint requirements planning sessions can be summarised as follows:

- Inaccurate translations – The overload of work that translators receive has a negative influence on the quality of translations.
- A system that contains a memory of terms previously coined and used in similar documents is not available. The NLS has a terminology-database, but this is not electronically accessible by the translators due to proprietary licensing restrictions. The translators do however receive a hard copy of the terminology database.
- In the past, some translators used SDL Trados (see www.trados.com), a translation memory (TM) system. SDL Trados is not used anymore, due to software compatibility and licensing issues.
- No online machine translation systems exist for South African languages. In general, insufficient online resources are available for African languages.
- Translators do not use a standard file naming convention. The consequence of this is difficulty in finding parallel translations of the same document.
- Hardware – Most translators do not have dual monitors connected to their computers. Dual monitors can be useful when translating between electronic documents.
- Translators make no use of previously translated documents. Some documents contain duplicate information that can be copied instead of translated again. Previously translated documents can also act as a template for new documents on the same topic or genre.
- Some translators perform translation by overtyping on the original document. They do not keep copies of the original documents, making it impossible to obtain parallel-translated documents.
- The South African government have decided to implement open source software in all government departments. Proprietary operating systems and word processors are being phased out in favour of open source solutions. The problem with this is that the change to open source is happening very slowly. Some departments have already implemented open source, while others are still using proprietary software. The implication of this is that all software developed for the

South African government must be open source and cross-platform compatible.

- Most translators use Microsoft® Office Word as their default translation environment. They are used to the graphic user interface of Microsoft® Office Word, and therefore would prefer a computer assisted translation program with a similar user interface to that of Microsoft® Office Word.
- Translators working at government departments have basic computer skills. Translators require computer assisted translation tools with a shallow learning curve that are simple and intuitive to use.
- Surprisingly, most translators are not opposed to machine translation systems.

The abovementioned preferences were analysed and rendered into end-user requirements, which steered the design and development of *Autshumato* ITE.

3 Related Work

Translation memories and machine translation systems are traditionally seen as opposing technologies. Although combining translation memory with machine translation is not a novel idea (Mügge, 2001 and Shuttleworth, 2002), very few computer-assisted translation tools exist that incorporate both technologies. We have an active interest in machine translation, and believe that both machine translation and translation memory have an important role to play in the translation process, especially in cases of languages with limited resources. We therefore decided to incorporate both technologies in the *Autshumato* ITE.

Other computer-assisted translation (CAT) applications that employ both machine translation and translation memory include the SDL Knowledge-based Translation System™ (http://www.sdl.com), Lingo24 ContexTrans (www.lingo24.com) and the ESTeam Translator© (www.esteam.se).

4 Design and Implementation

4.1 Integrated Solution

The project requirements (see Section 2) indicate the need for an easy and effective integrated translation environment for translators with basic computer skills. The term integrated refers to the fact that all the translation tools/resources required by the translator are accessible from within a single environment. These translation tools and resources include translation memory, machine translation, and term banks (glossaries). A diagram of the tools/resources that provide input to the ITE is displayed in Figure 1. The ITE must furthermore support open standards (i.e. TMX, XLIFF, etc.) to ensure compatibility with other translation tools. Using open standards also ensures that information is always accessible, and never becomes locked away within a proprietary format requiring a legacy application to access.

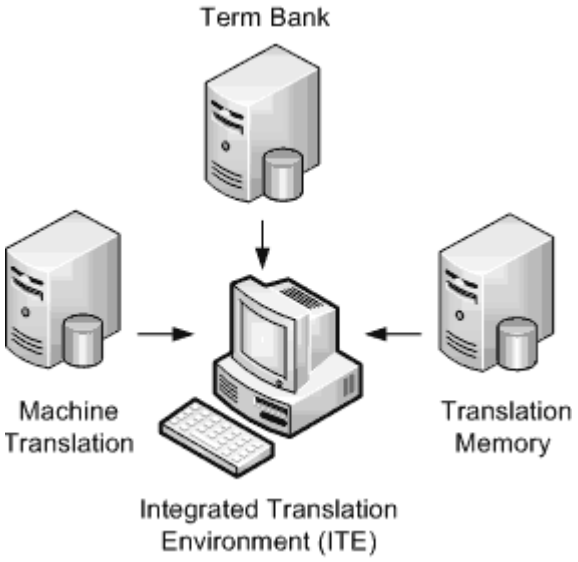


Figure 1. Translation Tools incorporated in the ITE.

4.2 Translation Workflow

Another important requirement is that ITE must be streamlined to provide a simple linear translation workflow. By following the linear flow: *Open* → *View* → *Translate* → *Edit* → *Restart*, we ensure that the translation of a document proceeds along a logical path. A diagram of the linear translation workflow is displayed in Figure 2.

4.3 Open Source Components

Since *Autshumato* is an open source project, we are utilising various existing open source components in the ITE. The main components used are the OpenOffice.org office suite, the OmegaT® CAT tool and the Moses Statistical Machine Translation system.

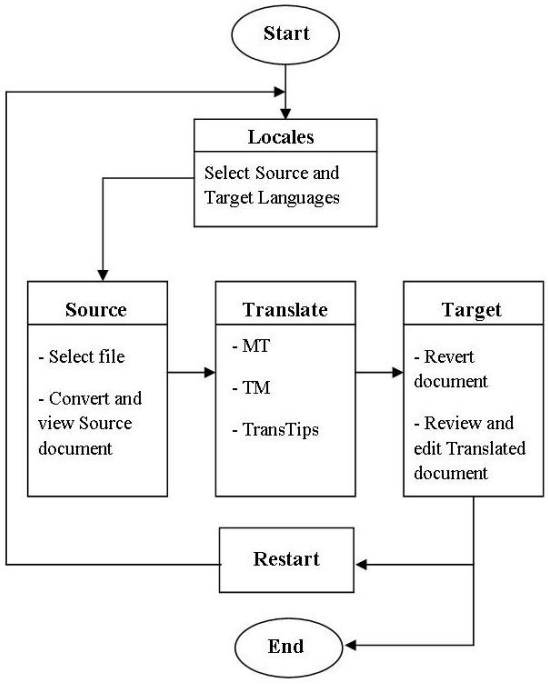


Figure 2. Workflow diagram.

OpenOffice.org is a popular office suite used for word processing, databases, spreadsheets and more. It can open documents from various other office packages and works on all the main operating systems and computers. It delivers documents in the international open standard and is open-source, which is free to everybody (www.openoffice.org). We use OpenOffice.org for its open document standard and relative ease in incorporating into other programs. Another important motivation for using OpenOffice.org is its resemblance to Microsoft® Office, since the translators indicated their preference for a translation environment with a similar interface to and all the functionalities of Microsoft® Office Word, during the end-user requirements elicitation.

OmegaT® is a popular CAT tool, which enables translators to effectively utilise TM, fuzzy matching and term banks to assist in the translation process (www.omegat.org). It already incorporates a large number of the functionalities required for *Autshumato* ITE, and is a very active open source project.

Moses is a statistical machine translation system using factored phrase-based beam-search decoding to deliver in-time translations. It has to be trained for every language pair with a collection of parallel corpora (Koehn et al, 2007). The Moses statistical machine translation system was chosen because it is currently one of the most advanced open source statistical machine translation systems.

The ITE is being developed with the Java programming language, a popular programming language for open source projects. Java was chosen to ensure compatibility and easy integration with other open source projects. The ITE is being primarily designed specifically for translators translating between the eleven South African languages; it can however be easily adapted for translating between any language pair in the world.

4.4 User Interface

In order for the ITE to appear simple and intuitive to use, the user interface should be logical and foreseeable. It is imperative that users do not get confused with endless options and functions. Functions that are more frequently used (like TM, MT, and TransTips) are automated to provide simpler interaction. The rest of the most commonly used options and functions are present on the toolbar. Usability is also improved by specifying the highlight colours for the source and translated texts, making it easier to identify untranslated sections.

The original design of the graphic user interface (GUI) entailed a split screen approach. This approach caused the GUI to be vertically split into two equally sized parts, with the source text being displayed on the left and the target text on the right. Translators disapproved of the split screen idea; they were concerned that the split screen would result in smaller working areas, which would in turn strain their eyes. We agreed that the split screen would not be practical, especially when working with an older desktop computer or notebook with a small monitor. We decided that a better alternative to the split screen approach would be to display the source and the target texts in separate windows, with the actual translation taking place in a third window.

The three windows are respectively named “Source”, “Translate” and “Target”. The three windows conform to the “View”, “Translate” and “Edit” phases in the unidirectional translation workflow. More information on the three windows is provided in the next section.

5 Main Functionalities

5.1 Source Window

The source window displays a read-only copy of the source document in its original form. The

translator is prevented from editing the source document to ensure true parallel target translations in the event of a single source document being translated into more than one target language. The source document serves only as a reference for comparing the target document during the translation process. The source window contains an embedded OpenOffice.org Writer for displaying the source document.

The source document is automatically converted to OpenOffice.org Writer format, without losing any text, pictures, tables, graphs etc. The source window supports the following file formats: Microsoft® Office Word (*.doc), OpenOffice.org Writer (*.odt), Xml (*.xml) and Html (*.html). Other formats like OpenDocument Spreadsheet (*.ods), OpenDocument Presentation (*.odp), and Office Open XML are to be included in future versions. Figure 2 shows an example of a document being displayed in the source window.

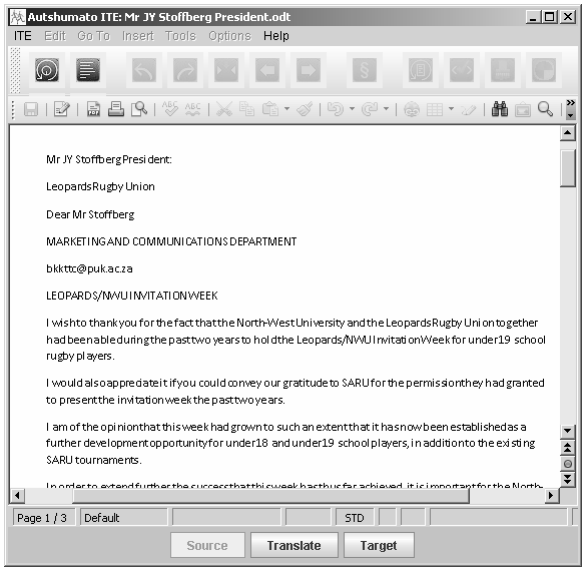


Figure 2. The Source window.

5.2 Translation Window

The Translation Window is where the actual translation is performed and consists of an embedded (and modified) OmegaT® CAT tool. The window is split into three panels: Translate, Fuzzy Matches and Machine Translation. A screenshot of the translate window is showed in Figure 3.

The translate panel displays segments of the source document for translation. Two choices of segmentation, namely paragraph and sentence segmentation are available to the user. Every segment contains a space where the translation of the involved segment must be inserted. The translator translates the entire source document in a segment-by-segment fashion in the translation window.

Various procedures work in the background to support the translator in creating an accurate translation. When a segment is activated, the program searches the TM for possible matches and displays the five best matches in the fuzzy match panel. The segments are matched by determining the Levenshtein distance (Levenshtein, 1965) between the sentence or paragraph in the involved segment and the translations contained in TM. All newly translated segments are constantly included in the TM, which greatly speeds up the translation of documents in the same context or domain.

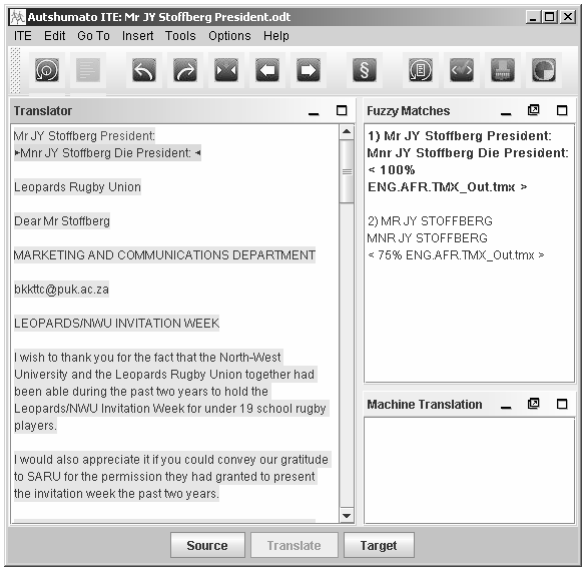


Figure 3. The Translation window.

Every word in the segment is matched against the glossary for finding word translations. Words for which matches are found are displayed in a different colour than the rest of the text in the segment. If the translator hovers the cursor over a matched word, the translation of the particular word is displayed in a pop-up window. These word translations are called TransTips. Figure 4 shows an example of a TransTip displaying the word “uitnodiging”, which is the Afrikaans translation of “invitation”. The applicable TM and glossary are loaded according to the source and target languages specified at the beginning of the translation workflow.

Upon entering the translation window, the entire source document is submitted to a server

running the Moses machine translation system. After the document has been translated by the server, it is downloaded to the ITE. The machine translation of every segment is displayed in the Machine Translation panel.

The translator can specify the editing behaviour of the segments. This provides the options of leaving the translate segment empty, automatically inserting the best fuzzy match, automatically inserting the machine translation, or simply copying the source text for overtyping. The translator can also combine fuzzy matches and machine translations by copying translated phrases from the Fuzzy Matches and MT panels.

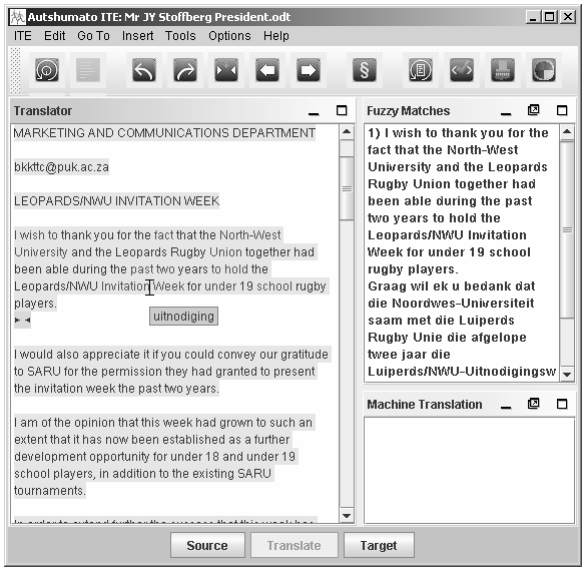


Figure 4. A TransTip being displayed.

5.3 Target Window

The target window displays the translated document in another embedded OpenOffice.org Writer component. It is fully editable and contains most of the OpenOffice.org Writer word processing functionalities.

Translators working for Government departments are required to produce translated documents with the same formatting as the original source documents. For this reason, the text formatting (style, font, size, colour, etc.) of the source document is saved in a skeleton file before the document is imported into the ITE. This formatting is automatically applied to the newly translated document to ensure it looks exactly the same as the original.

In cases where a word (or phrase) in a source sentence contains formatting different from the rest of the sentence, it is not always possible for the ITE to automatically apply the formatting to the correct target word, especially when the target word is not contained in the glossary. To overcome this problem we display tags in the source text to indicate formatting in the translation window. The tags usually appear in pairs, with rare occurrences of singular tags. The translator is required to carry over the correct formatting to the target text by using the Tag Painter tool. The ITE warns the translator when a segment contains source text formatting that has not been carried over to the target text. The advantage of the Tag Painter Tool is that the translator spends less time formatting and more time on translation.

This translated document in the target window can be compared to the original document in the source window. The translator is forced to save his/her work using a standard file naming convention, this is to prevent instances where the source document and the translated document cannot be linked. The target document can be saved in any of the following output formats, OpenDocument text (*.odt), Microsoft® Office Word (*.doc) or Portable Document Format (*.pdf). Figure 5 shows the translated document in the target window.

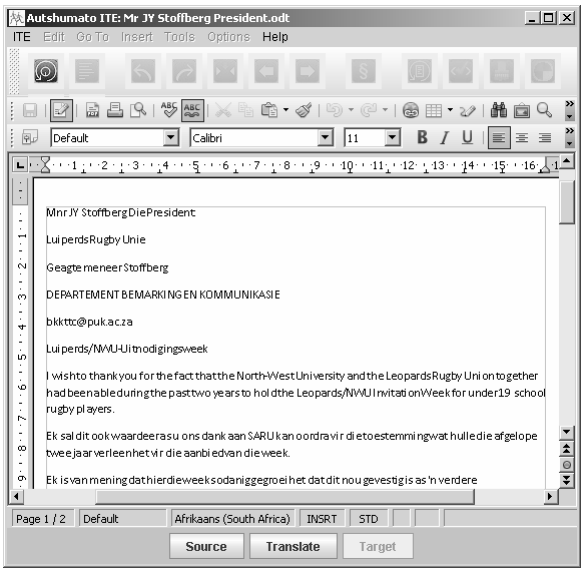


Figure 5. The Target window.

6 Conclusion

In this paper, we have introduced *Autshumato* ITE, a free and open source integrated translation environment that facilitates a variety of translation tools and resources. An important

feature of *Autshumato* ITE is that it offers both machine translation and translation memory to help translators to improve the quality and quantity of their translations.

Future work includes extending *Autshumato* ITE's capability to incorporate a fully-fledged terminology management system that will be accessible to both translators and terminologists. We are also interested in incorporating a document management system. The ITE currently supports only a limited number of file formats and we want to extend the list of supported file formats for input and output considerably .

We have vast experience in the development of spelling checkers for South African languages and want to apply this knowledge to create open source spelling checkers that can be implemented into the ITE. Other functionalities and improvements will be made based on the feedback we expect to receive from the users of *Autshumato* ITE.

We are also creating a website that will host an open source community for the *Autshumato* project. The website will contain downloadable resources, source code and complete binary packages for deployment.

One of the biggest challenges we are facing, is the development of machine translation systems for the official South African languages. Machine translation systems require large sets of parallel corpora and as previously mentioned in this article, this is a very scarce resource for South African languages. We are in the process of gathering parallel data for developing and improving existing machine translation systems for South African languages. We are aware of the fact that statistical machine translation might not be the ideal approach for creating machine translation systems for resource-scarce languages with small amounts of parallel corpora. For this reason where are doing research on alternative ways than mere addition of parallel data, to improve the output of the statistical machine translation systems. We do however believe that *Autshumato* ITE will contribute in generating more parallel data by stimulating translation to resource-scarce languages. In turn, this would help us to improve the quality of statistical machine translations for these languages.

References

Giliomee, Herman and Mbenga, Bernard. 2007. *New History of South Africa*, Tafelberg, Cape Town.

Koehn, Philipp., Hoang, Hieu., Birch, Alexandra., Callison-Burch, Chris., Federico, Marcello., Bertoldi, Nicola., Cowan, Brooke., Shen, Wade., Moran, Christine., Zens, Richard., Dyer, Chris., Bojar, Ondrej., Constantin, Alexandra., and Herbst, Evan. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demonstration and Poster Session*. Prague, Czech Republic. 177-180

Levenshtein, Vladimir I. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10:707-710.

Mügge, Uwe. 2001. The Best of Two Worlds - Integrating Machine Translation into Translation Memory Systems: A universal approach based on the TMX standard. *Language International*, 13 (6): 26-29.

Republic of South Africa, 1996. *Constitution of the Republic of South Africa,* Act 108 of 1996. Government Printer, Pretoria.

Shuttleworth, Mark. 2002. Combining MT and TM on a Technology-oriented Translation Masters: Aims and Perspectives. Proceedings of the *6th BCS EAMT Workshop Teaching Machine Translation*. Manchester, England.

Van der Merwe, I.J. and Van der Merwe, J.H. 2006. *Linguistics Atlas of South Africa*, SUN PReSS, Stellenbosch.

A New Subtree-Transfer Approach to Syntax-Based Reordering for Statistical Machine Translation

Maxim Khalilov and **José A.R. Fonollosa**
Universitat Politècnica de Catalunya
Campus Nord UPC, 08034,
Barcelona, Spain
{khalilov,adrian}@gps.tsc.upc.edu

Mark Dras
Macquarie University
North Ryde NSW 2109,
Sydney, Australia
madras@ics.mq.edu.au

Abstract

In this paper we address the problem of translating between languages with word order disparity. The idea of augmenting statistical machine translation (SMT) by using a syntax-based reordering step prior to translation, proposed in recent years, has been quite successful in improving translation quality. We present a new technique for extracting syntax-based reordering rules, which are derived through a syntactically augmented alignment of source and target texts. The parallel corpus with reordered source side is then passed to an N-gram-based machine translation system and the obtained results are contrasted with a monotone system performance. In experiments, we show significant improvement for the Chinese-to-English translation task.

1 Introduction

One of the most challenging problems facing machine translation (MT) is how to place the translated words in the natural order of the target language. A monotone SMT system suffers from weakness in the distortion model, even if it is able to generate correct word-by-word translation. In this study we propose a reordering model that involves both source- and target-side syntax information in the word reordering process.

Our work is inspired by the approach proposed in Imamura et al. (2005), where a complete syntax-driven SMT system based on a two-side subtree transfer is described. In their approach they construct a probabilistic non-isomorphic tree mapping model based on a context-free breakdown of the source and target parse trees; extract alignment templates that incorporate the constraints of the parse trees; and apply syntax-based decoding. We propose to use a similar non-isomorphic subtree mapping to extract reordering rules, but instead of involving them directly in the translation process, we use them to monotonize the source portion of the bilingual corpus.

In the next step, the rules are applied to the source part of the same training corpus changing the source sentence structure such that it more closely matches the word order of the target language. It leads to a simplification of the translation task due to a shorter average length of bilingual units which it is more likely to see when translating an unseen set.

Local and long-range word reorderings are driven by automatically extracted permutation patterns operating with source language constituents and underlaid by non-isomorphic subtree transfer. The target-side parse tree utilization is optional, but it greatly affects system performance: it is considered as a filter constraining the reordering rules to the set of patterns covered by both the source- and target-side subtrees. Apart from the reordering rules representing the order of child nodes, a set of additional rewrite rules based on a deep top-down subtree analysis is considered, which is another novel aspect of the paper.

We used the N-gram-based SMT system of Mariño et al. (2006) to test the proposed syntax-based reordering model, which is an alternative to the phrase-based state-of-the-art Moses[1] system.

2 Related work

In practice, a reordering model operates on a sentence level and is carried out based on word reordering rules derived from the training corpus. Reordering patterns can be purely statistical (see Costa-jussà and Fonollosa (2006), for example), use language-based syntactic information (Collins et al., 2005); the reordering can be driven by a lat-

[1] www.statmt.org/moses/

Proceedings of the 13th Annual Conference of the EAMT, pages 197–204,
Barcelona, May 2009

tice of syntactically motivated alternative translations (Elming, 2008) or be based on automatically extracted patterns driven by syntactical structure of the languages (see Crego and Mariño (2007b) as an example). Another recent implementation of the preprocessing approach to syntax-based reordering though an n-best list generation can be found in Li et al. (2007).

Word class-based reordering patterns were part of Och's Alignment Template system (Och et al., 2004). The modern state-of-the-art phrase-based translation system Moses, along with a distance based distortion model (Koehn et al., 2003), implements the phrase-based reordering (Tillmann and Zhang, 2005).

Reordering algorithms specifically developed for an N-gram system include a constrained distance-based distortion model (Costa-jussà et al., 2006) and a linguistically motivated reordering model employing monotonic search graph extension (Crego and Mariño, 2007a).

An example of a word order monotonization strategy can be found in Costa-jussà and Fonollosa (2006), where a monotone sequence of source words is translated into the reordered sequence using SMT techniques.

In Xia and Mccord (2004) the authors present a hybrid system for French-English translation, based on the principle of automatic rewrite patterns extraction using a parse tree and phrase alignments. This method differs from the one presented in this paper, among other distinctions, by a lexical model underlying the subtree syntax transfer and a different statistical model used for translation.

3 Baseline SMT system

N-gram-based SMT has proved to be competitive with the state-of-the-art systems in recent evaluation campaigns (Khalilov et al., 2008; Lambert et al., 2007).

According to the N-gram-based approach, the translation process is considered as an *arg max* searching for the translation hypothesis $\hat{e}_1^I$ maximizing a log-linear combination of a translation model (TM) and a set of feature models:

$$\hat{e}_1^I = \arg\max_{e_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \quad (1)$$

where the feature functions h_m refer to the system models and the set of λ_m refers to the weights corresponding to these models.

A detailed description of the N-gram-based approach can be found in Mariño et al. (2006).

As decoder, we used MARIE[2] (Crego et al., 2005), a beam-search decoder implementing a distance-based constrained distortion model, limited by two parameters: *m* - a maximum distance measured number in words that a phrase can be reordered and *j* - a maximum number of "jumps" within a sentence (Costa-jussà et al., 2006).

4 Syntax-based reordering

Our syntax-based reordering (SBR) system requires access to source and target language parse trees, along with the source-to-target and target-to-source word alignments intersection. In the framework of the study we used the Stanford Parser (Klein and Manning, 2003) for both languages, however the system permits using any other natural language parser allowing for different formal grammars for the source and the target languages.

4.1 Notation

SBR operates with source and target parse trees that represent the syntactic structure of a string in source and target languages in a Context-Free Grammar (CFG) fashion.

This representation is called *"CFG form"*, and is formally defined in the usual way as $G = \langle N, T, R, S \rangle$, where N is a set of nonterminal symbols (corresponding to source-side phrase and part-of-speech tags); T is a set of source-side terminals (the lexicon); R is a set of production rules of the form $\eta \rightarrow \gamma$, with $\eta \in N$ and γ a sequence of terminal and nonterminal symbols; and $S \in N$ is the distinguished symbol.

The reordering rules then have the form

$$\eta_0 @0 \ldots \eta_k @k \rightarrow \eta_{d_0} @d_0 \ldots \eta_{d_k} @d_k |Lexicon| p_1 \quad (2)$$

where $\eta_i \in N$ for all $0 \leq i \leq k$; $(d_o \ldots d_k)$ is a permutation of $(0 \ldots k)$; *Lexicon* includes the source-side set of words for each η_i; and p_1 is a probability associated with the rule. Figure 1 gives two examples of the rule format.

4.2 Rule extraction

Concept. Inspired by the ideas presented in Imamura et al. (2005), where monolingual correspon-

[2] http://gps-tsc.upc.es/veu/soft/soft/marie/

dences of syntactic nodes are used during decoding, we extract a set of bilingual patterns allowing for reordering as described below:

(1) align the monotone bilingual corpus with GIZA++[3] (Och and Ney, 2003) and find the intersection of direct and inverse word alignments, resulting in the construction of the projection matrix P (see below);

(2) parse the source and the target parts of the parallel corpus;

(3) extract reordering patterns from the parallel non-isomorphic CFG-trees based on the word alignment intersection.

Step 2 is straightforward; we explain aspects of Steps 1 and 3 in more detail below. Figure 1 shows an example of the generation of two lexicalized rules; we use this below in our explanations.

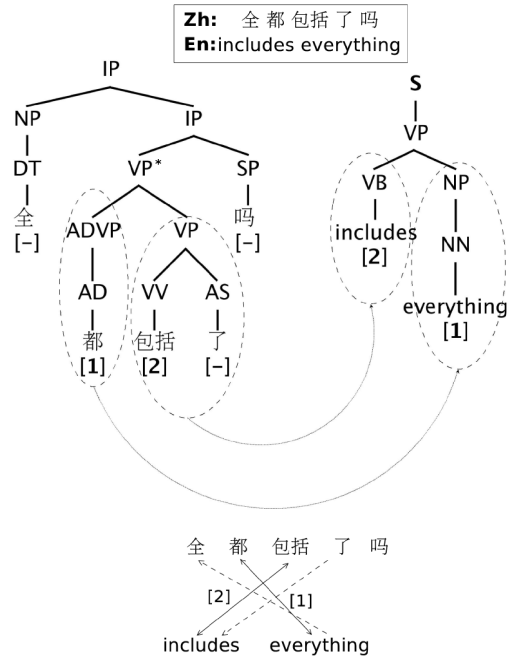


Figure 1: Example of reordering rules extraction.

Projection matrix. Bilingual content can be represented in the form of words or sequences of words depending on the syntactic role of the corresponding grammatical element (constituent or POS).

[3]http://code.google.com/p/giza-pp/

Given two parse trees and a word alignment intersection, a projection matrix P is defined as an $M \times N$ matrix such that M is the number of words in the target phrase; N is the number of words in the source phrase; and a cell (i, j) has a value based on the alignment intersection — this value is zero if word i and word j do not align, and is a unique non-zero link number if they do.

For the trees in Figure 1,

$$P = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Alignment and sub-trees interaction. Each non-terminal from the source and target parse trees is assigned a string carrying information about elements from the alignment intersection which are contained in its child nodes, taking into account the order of their appearance in the tree (AI). For example, the AI string assigned to the source-side internal node VP^* in Figure 1 is "1 2" and to the target-side VP is "2 1". This information is used to indicate the source-side nodes which are to be reordered according to the target language syntactical structure. Reordering patterns are extracted following the source and target-side AIs as shown in Figure 1 ("*main rules*").

If more than one non-zero element of the projection matrix is reachable through the child nodes, the AI has a more complex structure, providing information about elements from alignment intersection belonging to one or another child node. An example can be found in Figure 2.

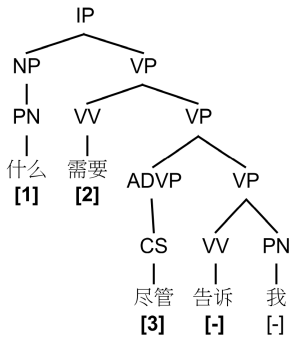


Figure 2: Example of complex AI structure.

Here, the subtree IP is assigned with the AI_{IP} = "1 (2 3)", meaning that it has two child nodes: the first contains the element 1 from the alignment intersection and the second, elements 2 and 3 (we call this subsequence "*closed*"). The reordering system considers nodes assigned with one or more

children equally discerning the nodes with different order alignment elements.

Unary chains. Given a unary chain like "$ADVP \rightarrow AD \rightarrow ...$", rules are extracted for each level in this chain. For example in Figure 1, the directly extracted reordering rules are equivalent since the node $ADVP$ leads to the leaf through the node AD and does not have other edges.

The role of target-side parse tree. Conceptually speaking, the use of target-side parse tree is optional. Although reordering is performed on the source side only, the target-side tree is of great importance: the reordering rules can be only extracted if the words covered by the rule are entirely covered by both a node in the source and in the target trees. It allows the more accurate determination of the covering and limits of the rules.

4.3 Secondary rules

There are a lot of nodes for which a comparison of AIs indicates that a subtree transfer can be done, but segmentation of child nodes is not identical.

Figure 3 illustrates this situation. AI strings assigned to the root nodes of the trees contain the same elements, but segmentation and/or order of appearance of elements do not coincide. These subtrees can not be directly used for pattern extraction and more in-depth analysis is required.

We adopt the following six step algorithm for each parent node from the source-side parse tree:

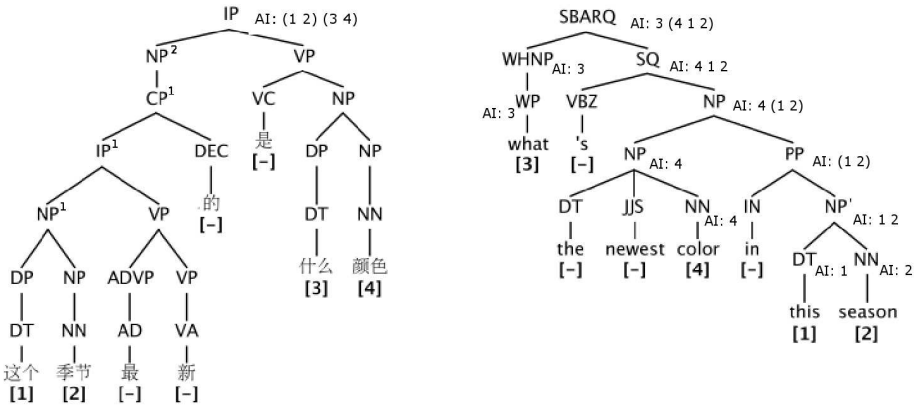
1. Find the AI sequence for the source-side top-level element (considering example, IP node is assigned "(1 2) (3 4)").

2. Go down through the target-side tree, finding AIs for each node.

3. Find all target-side closed subsequences for the source-side AI found on step 1. In example, it is the subsequence "(1 2)".

4. Find all target-side isolated nodes corresponding to the elements which were not covered on step 2. In example, these elements are "3" and "4".

5. Extend the set of source-side nodes found in steps 2 and 3 with equivalent branches. Since the words which are not presented in the alignment intersection do not affect the projection matrix, "equivalence" means here that all the branches spanning the elements from



Example of extracted rules:

NP@0 DP@1 NP@2 -> DP@1 NP@2 NP@0 | NP@ << 这个 季节 >> DP@1 << 什么 >> NP@2 << 颜色 >>
NP@0 DP@1 NN@2 -> DP@1 NN@2 NP@0 | NP@ << 这个 季节 >> DP@1 << 什么 >> NN@2 << 颜色 >>
NP@0 DT@1 NP@2 -> DT@1 NP@2 NP@0 | NP@ << 这个 季节 >> DT@1 << 什么 >> NP@2 << 颜色 >>
NP@0 DT@1 NN@2 -> DT@1 NN@2 NP@0 | NP@ << 这个 季节 >> DP@1 << 什么 >> NN@2 << 颜色 >>
CP@0 DP@1 NP@2 -> DP@1 NP@2 CP@0 | CP@0 << 这个 季节 最 新 的 >> DP@1 << 什么 >> NP@2 << 颜色 >>
CP@0 DP@1 NN@2 -> DP@1 NN@2 CP@0 | CP@0 << 这个 季节 最 新 的 >> DP@1 << 什么 >> NN@2 << 颜色>>
CP@0 DT@1 NP@2 -> DT@1 NP@2 CP@0 | CP@0 << 这个 季节 最 新 的 >> DT@1 << 什么 >> NP@2 << 颜色 >>
CP@0 DT@1 NN@2 -> DT@1 NN@2 CP@0 | CP@0 << 这个 季节 最 新 的 >> DT@1 << 什么 >> NN@2 << 颜色>>
NP@0 DP@1 NP@2 -> DP@1 NP@2 NP@0 | NP@0 << 这个 季节 最 新 的 >> DT@1 << 什么 >> NN@2 << 颜色>>
NP@0 DP@1 NN@2 -> DP@1 NN@2 NP@0 | NP@0 << 这个 季节 最 新 的 >> DT@1 << 什么 >> NN@2 << 颜色>>
NP@0 DT@1 NP@2 -> DT@1 NP@2 NP@0 | NP@0 << 这个 季节 最 新 的 >> DT@1 << 什么 >> NN@2 << 颜色>>
NP@0 DT@1 NN@2 -> DT@1 NN@2 NP@0 | NP@0 << 这个 季节 最 新 的 >> DT@1 << 什么 >> NN@2 << 颜色>>
...
NP@0 VP@1 -> VP@1 NP@0 | NP@0 << 这个 季节 最 新 的 >> VP@1 << 是 什么 颜色 >>

Figure 3: Example of “secondary“ rules extraction.

the given instance are considered equally (for example, elements NP^1 are equivalent to the nodes IP^1, CP^1 and NP^2).

6. Place them in order corresponding to the target-side AI and construct the final reordering patterns (*"secondary rules"*).

As illustration of the limitations incurred by target-side parse tree, the potential reordering pattern $NP@0\ VP@1 \rightarrow VP@1\ NP@0$ (referring to the top node in the Chinese tree) is not allowed due to distinct source- and target-side tree coverage.

4.4 Rule organization

Once the list of fully lexicalized reordering patterns is extracted, all the rules are progressively processed, reducing amount of lexical information. Initial rules are iteratively expanded such that each element of the pattern is generalized until all the lexical elements of the rule are represented in the form of fully unlexicalized categories. Hence, from each initial pattern with N lexical elements, $2^N - 2$ partially lexicalized rules and 1 general rule are generated. An example of the process of delexicalization can be found in Figure 4.

Thus, finally three types of rules are available: (1) fully lexicalized (initial) rules, (2) partially lexicalized rules and (3) unlexicalized (general) rules.

On the next step, the sets are processed separately: patterns are pruned and ambiguous rules are removed. Fully and partially lexicalized rules are not pruned out, but we set the threshold k_{gener} to 3. All the rules from the corresponding set that appear less than k times are directly discarded. The probability of a pattern is estimated based on frequency in the training corpus, and only the most probable rule is stored.

In this version of the reordering system, only the one-best reordering is used in other stages of the algorithm, so the rule output functioning as an input to the next rule can lead to situations reverting the change of word order that the previously applied rule made. Therefore, the rules that can be ambiguous when applied sequentially are pruned according to the higher probability principle. For example, for the pair of patterns with the same lexicon (which is empty for a general rule leading to a recurring contradiction *NP@0 VP@1 → VP@1 NP@0 p1*, *VP@0 NP@1 → NP@1 VP@0 p2*), the less probable rule is removed.

Finally, there are three resulting parameter tables analogous to the "r-table" as stated in (Yamada and Knight, 2001), consisting of POS- and constituent-based patterns allowing for reordering and monotone distortion.

4.5 Source-side monotonization

Rule application is performed as a bottom-up parse tree traversal following two principles:

(1) the longest possible rule is applied, i.e. among a set of nested rules, the rule with a longest left-side covering is selected. For example, in the case of the appearance of an *NN JJ RB* sequence and presence of the two reordering rules

$$NN@0\ JJ@1 \rightarrow \ldots \text{ and}$$

$$NN@0\ JJ@1\ RB@2 \rightarrow \ldots$$

the latter pattern will be applied.

(2) the rule containing the maximum lexical information is applied, i.e. in case there is more than one alternative pattern from different groups, the lexicalized rules have preference over the partially lexicalized, and partially lexicalized over general ones.

Figure 5 shows example of the reordered source-side tree corresponding to the example from Fig-

Initial rule:
QP@0 CP@1 NP@2 -> QP@0 CP@1 NP@2 | QP0@ << 个 >>CP1@ << 不错 >>NP2@ << 夜总会 >>

Partially lexicalized rules:
QP@0 CP@1 NP@2 -> QP@0 CP@1 NP@2 | QP@0 << 个 >>CP@1 << NON >> NP@2 << NON >>
QP@0 CP@1 NP@2 -> QP@0 CP@1 NP@2 | QP@0 << NON >>CP@1 << 不错 >> NP@2 << NON >>
QP@0 CP@1 NP@2 -> QP@0 CP@1 NP@2 | QP@0 << NON >>CP@1 << NON >> NP@2 << 夜总会 >>
QP@0 CP@1 NP@2 -> QP@0 CP@1 NP@2 | QP@0 << NON >>CP@1 << 不错 >> NP@2 << 夜总会 >>
QP@0 CP@1 NP@2 -> QP@0 CP@1 NP@2 | QP@0 << 个 >>CP@1 << NON >> NP@2 << 夜总会 >>
QP@0 CP@1 NP@2 -> QP@0 CP@1 NP@2 | QP@0 << 个 >>CP1@ << 不错 >> NP@2 << NON >>

General rule:
QP@0 CP@1 NP@2 -> QP@0 CP@1 NP@2

Figure 4: Example of lexical rule expansion.

ure 1 with the applied pattern

$$ADVP@0\ VP@1 \rightarrow VP@1\ ADVP@0$$

and the given lexicon. The resulting reordered Chinese phrase more closely matches the order of the target language and is considered as a result of the subtree transfer.

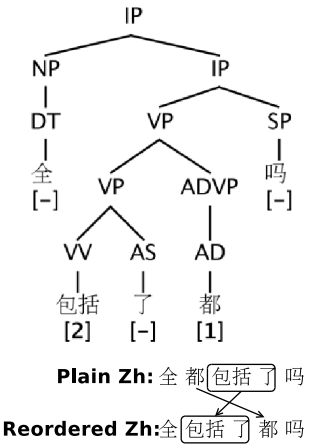


Figure 5: *Reordered source-side parse tree.*

Once the reordering of the training corpus is ready, it is realigned and new more monotonic alignment is passed to the SMT system. In theory, the word links from the original alignment can be used, however, from our experience, running GIZA++ again results in a better word alignment since it is easier to learn on the modified training example.

5 Experiments and results

5.1 Corpus

The experiments were conducted on two Chinese-English corpora: the BTEC corpus consisting of short tourism related sentences and the 50K first-lines extraction from the NIST'06 corpus belonging to the news domain (NIST50K). The main reason why the Chinese-English translation task was chosen for experiments is that European languages are not so crucial for global (long-distance) reordering problem as the translation between Asian languages and English.

We expect that the need for longer distance reorderings would be found in longer sentences, as in the NIST50K corpus, but we also include the BTEC corpus to see whether there is an effect for shorter sentences as well. Basic statistics of the training material can be found in Tables1 and 2.

Both system were optimized and tested on in-domain data. BTEC development and test datasets consist of 489 and 500 sentences, respectively, and are provided with 7 reference translations. NIST50K development and test sets are both 541 sentences long, 4 references are provided.

5.2 Experiment setup

Evaluation conditions were case-insensitive and with punctuation marks considered. We used the Stanford Parser as a NLP parsing engine (Klein and Manning, 2003) trained on the Chinese and English Penn Treebank sets (32 POS/44 constituent categories for Arabic Treebank and 48 POS/14 syntactic tags for English Treebank).

N-gram models were estimated using the SRILM toolkit (Stolcke, 2002). For both tasks TM is represented in a 4-gram model form using modified Kneser-Ney discounting with interpolation, target language model (LM) of words is a 4-gram model with modified Kneser-Ney discounting, while a target-side POS LM is a 4-gram with Good-Turing backing-off.

For all system configurations, apart from monotone experiments, parameters of the distance-based reordering model were set to $m = 5$ and $j = 5$ for a trade-off between efficiency and accuracy.

The optimization criteria was the highest $4NIST + 100BLEU$ score.

5.3 Results and discussion

The following scores are reported in Table 3: final score obtained as a result of model weights tuning for development dataset (*dev*), BLEU and METEOR scores for the test dataset (*test*). We present results for two corpora: BTEC and NIST05K characterized by different domain and sentence length.

We contrast four n-gram-based system configurations comparing the SBR results with the distortion model:

- *Baseline*: the training data is not reordered and allows for *C*onstrained *D*istortion ($m = 5, j = 5$) during decoding, as described in (Costa-jussà et al., 2006);
- *SynBReor*: SBR is applied on the preprocessing step involving *main rules* only, the dev/test sets are monotonically decoded;
- *SynBReor+SecRules*: SBR is applied involving *main* and *secondary rules* and allows for constrained distortion ($m = 5, j = 5$).

	Chinese	English
Sentences	44.9 K	44.9 K
Words	299.0 K	324.4 K
Average sentence length	6.66	7.22
Vocabulary	11.4 K	9 K

Table 1: Basic statistics of the BTEC training corpus.

	Chinese	English
Sentences	50 K	50 K
Words	1.18 M	1.25 M
Average sentence length	23.6	25.03
Vocabulary	27.2 K	30.4 K

Table 2: Basic statistics of the NIST50K training corpus.

Application of the SBR technique demonstrates an improvement in translation quality according to the automatic scores. *SynBReor+mj* is found to be the best system configuration for both sets of experiments, outperforming the baseline configuration by about 0.4 BLEU points (2.9 %) that is not statistically significant for the BTEC task, however, for the NIST50K task the difference is about 0.9 BLEU points (4 %) reaching a statistical significance threshold[4]. The METEOR score also increases with raise of reordering system complexity, supporting the BLEU results. The SBR algorithm is illustrated in Figure 6, where the Chinese block of words is moved to the end of the sentence that better matches the structure of the English counterpart.

As usual, for the tasks with scarce resources the improvements on the *test* and *dev* sets are not coherent. While a clear improvement of test results can be observed in the BTEC results, the development set score degrades when SBR is applied.

It is possible to see from Table 3 that the introduction of secondary rules influences negatively the number of extracted tuples and comparing to the "main rules only" configuration shows a degradation in performance. Generally speaking, secondary rules include more elements than primary ones and are more difficult to be seen in the dataset parsed with the Stanford Parser. However, we speculate that accurate pruning of secondary rules could benefit the system performance significantly.

6 Conclusions and future work

In this paper we introduced a syntax-based reordering technique that monotonizes the word order of

[4]All statistical significance calculations are done for a 95% confidence interval and 1000 resamples (Koehn, 2004).

	dev	test BLEU	test METEOR	# tuples	voc tuples
BTEC experiments					
Baseline	48.17	19.50	47.05	150,378	36,643
SynBReor	47.55	19.91	47.50	157,345	36,936
SynBReor+SecRules	47.83	19.70	47.52	141,430	36,501
NIST50K experiments					
Baseline	19.16	21.28	41.55	240,609	112,947
SynBReor	19.90	22.21	41.77	252,113	114,702
SynBReor+SecRules	19.45	21.80	42.03	251,012	113,985

Table 3: Summary of the experimental results.

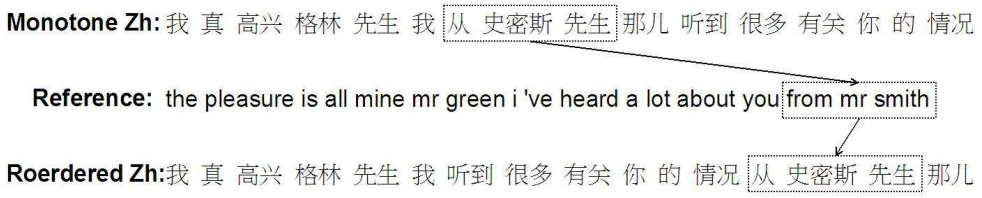


Figure 6: Example of SBR application.

source and target languages involved in the process of bilingual unit extraction. As can be seen from the results presented, the proposed algorithm shows competitive performance comparing with a fundamental distance-based reordering model.

The comparison is done on two smaller Chinese-English translation tasks with a strong need for word reorderings. The major part of the sentences from the BTEC corpus are short and on the example of the tourism translation task one can observe the SBR capacity to deal with local reordering. The NIST50K task demonstrates potential of the SBR algorithm on the translation task with much longer average sentence length and much need of long-distance reorderings. In this case, the reordered system significantly outperforms the state-of-the-art model.

The proposed approach is flexible and in the next step will be applied to phrase-based systems. Further work also includes the algorithm's application to a different language pair with distinct need for reorderings, analysis of the extracted tuples and development of the algorithm for accurate selection of reordering rules.

7 Acknowledgments

This work has been funded by the Spanish Government under grant TEC2006-13964-C03 (AVIVAVOZ project) and under a FPU grant.

References

Collins, M., Ph. Koehn, and I. Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proc. of the ACL'05*, pages 531–540.

Costa-jussà, M.R. and J.A.R. Fonollosa. 2006. Statistical machine reordering. In *Proc. of EMNLP'06*.

Costa-jussà, M.R., J. M. Crego, A. de Gispert, P. Lambert, M. Khalilov, J. A.R. Fonollosa, J. B. Mariño, and R. E. Banchs. 2006. TALP phrase-based system and TALP system combination for IWSLT 2006. In *Proc. of the IWSLT'06*, pages 123–129.

Crego, J.M. and J.B. Mariño. 2007a. Improving statistical MT by coupling reordering and decoding. *Machine Translation*, 20(3):199–215.

Crego, J.M. and J.B. Mariño. 2007b. Syntax-enhanced N-gram-based SMT. In *Proc. of MT SUMMIT XI*.

Crego, J.M., J.B. Mariño, and A. de Gispert. 2005. An ngram-based statistical machine translation decoder. In *Proc. of INTERSPEECH05*.

Elming, J. 2008. Syntactic reordering integrated with phrase-based SMT. In *Proc. of the ACL'08 Workshop SSST-2*, pages 46–54, June.

Imamura, K., H. Okuma, and E. Sumita. 2005. Practical approach to syntax-based statistical machine translation. In *Proc. of MT Summit X*.

Khalilov, M., A.H. Hernández, M.R. Costa-jussà, J.M. Crego, C.A. Henríquez, P. Lambert, J.A.R. Fonollosa, J.B. Mariño, and R. Banchs. 2008. The TALP-UPC ngram-based statistical machine translation system for ACL-WMT 2008. In *Proc. of WMT'08*, pages 127–131.

Klein, D. and C. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of the ACL'03*.

Koehn, Ph., F.J. Och, and D. Marcu. 2003. Statistical phrase-based machine translation. In *Proc. of the HLT-NAACL'03*, pages 48–54.

Koehn, Ph. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP'04*, pages 388–395.

Lambert, P., M.R. Costa-jussà, J. M. Crego, M. Khalilov, J. Mariño, R.E. Banchs, J.A.R. Fonollosa, and H. Shwenk. 2007. The TALP ngram-based SMT system for IWSLT 2007. In *Proc. of IWSLT'07*, pages 169–174.

Li, C., D. Zhang, M. Zhou, M. Li, and Y. Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proc. of ACL'07*, pages 720–727.

Mariño, J.B., 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. *Computational Linguistics*, 32(4):527–549.

Och, F.J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, F.J., 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. In *Proc. of HLT/NAACL'04*.

Stolcke, A. 2002. SRILM: an extensible language modeling toolkit. In *Proc. of ICSLP'02*.

Tillmann, C. and T. Zhang. 2005. A localized prediction model for statistical machine translation. In *Proc. of the ACL 2005*, pages 557–564.

Xia, F. and M. McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proc. of 2004*.

Yamada, K. and K. Knight. 2001. A syntax-based statistical translation model. In *Proc. of the ACL 2001*, pages 523–530.

On Extracting Multiword NP Terminology for MT

Svetlana Sheremetyeva
LanA Consulting ApS
Møllekrog, 4, Vejby
3210 Copenhagen, Denmark
lanaconsult@mail.dk

Abstract

The paper addresses the issue of MT knowledge acquisition and describes a new hybrid methodology for automatic extraction of multi-word nominal terminology. The approach is based on statistical techniques merged into a strongly lexicalized Constraint Grammar paradigm. It is targeted at intelligent output and computationally attractive properties.

1 Introduction

The quality of machine translation output is to a large extent influenced by the comprehensiveness of multiword term dictionaries where noun phrases (noun phrase terms) are more frequent than any other types of multiword expressions.

Noun phrases (NPs) can often be translated into other languages irrespective of the context and contribute significantly to the robustness of MT systems by reducing the ambiguity inherent in word to word matching and text analysis.

Multiword phrase databases are relevant for both RBMT and SMT systems; - in state-of-the-art statistical translation systems structural relations between source and target sentences are captured by means of phrases instead of isolated words (Zens et al., 2002; Koehn et al., 2003).

To build a two level hierarchy of phrases phrase driven SMT systems more and more focus on the linguistic concept of NP as the unit of decomposition (Hewavitharana et al, 2007).

Creating multilingual MT resources is based on the acquisition of unilingual lexicons as the first and basic step (Pohl, 2006; Hewavitharana et al, 2007; Daille and Morin, 2008).

In many cases, especially for low resource languages, lexical acquisition starts from the English side independent of the translation direction. For example, (Hewavitharana et al, 2007) developing an Arabic-to-English MT first extract English NPs as the Arabic parsers available do not produce desired accuracy. The quality of monolingual (English, in particular) extraction, is thus of primary importance for the quality of MT.

Another issue which matters a lot for practical MT systems is the speed of NP extraction process. It directly affects the costs of MT system development and maintenance. Despite a lot of research on extracting different kinds of multiword phrases related to MT the problem still presents a tough challenge (Piao et al., 2006). Extraction of NP phrases, in particular, is especially problematic as it normally involves parsing and often very expensive computationally.

We suggest a new hybrid NP extraction methodology based on statistical techniques merged into a strongly lexicalized Constraint Grammar paradigm which features computationally attractive properties and intelligent output. We illustrate our approach on the example of the English language as its resources are most widely used in MT research and the quality of these resources is still to be improved.

While testing our approach we got practical results for the patent domain corpus and saw how generic extraction procedures can take advantage of the domain restrictions. For patent MT the issue of extraction quality and speed is one of the priorities as its terminology is being constantly renewed and requires operative maintenance.

In what follows we first overview related work, we then define our task and describe the extraction procedure followed by evaluation results. We conclude with discussion and future work.

Proceedings of the 13th Annual Conference of the EAMT, pages 205–212,
Barcelona, May 2009

2 Related work

The range of the work related to NP term extraction is very wide and covers NP, multiword expression, collocation and keyphrase extraction. Keyphrase implies two features: phraseness and informativeness (Tomokiyo and Hurst, 2003) and while the issue of informativeness is beyond the scope of the current article, techniques used to identify phraseness are of direct interest to our research.

NP describes objects and concepts. It is a grammatical notion and techniques used for detecting noun phrases in the text are normally NLP-oriented. The most correct results in NP extraction can be expected with full-fledged NLP (symbolic) procedures, which while unquestionable under the assumption of perfect NLP parsing in reality will immediately lead to the problems of coverage, hence robustness and correctness[1]. Pure NLP parsing can be very time consuming and normally not portable.

An ultimate example of symbolic approach to extraction is a semantic tagger which annotates English corpora with semantic category information and is capable of detecting and semantically classifying many multiword expressions but can suffer from low recall (Rayson et al., 2004).

Current approaches to NP extraction in an attempt to raise recall and extraction speed involve statistical techniques where phrases, collocations or multiword expressions are determined as word sequences with no intention to limit the meaning in a linguistic sense. In pure statistical methods phrase extraction is based on n-gram extraction and may include such preprocessing steps as stoplist words removal and stemming[2]. Phrases are further selected based on various statistical collocation/phrasenessh metrics, e.g., mean and variance (Smadja, 1993) and binomial log likelihood ratio test (BLRT) (Dunning, 1993), to mention just a few.

On the one hand, statistical techniques offer some clear advantages, such as speed, robustness and portability, over linguistically-informed methods. On the other hand, the results obtained statistically are not always "good" phrases, and the basic statistical systems may suffer from combinatorial explosion if calculations are made over a large search space.

To overcome the limitations of "pure" approaches a use of statistics supplemented by heuristics and linguistic techniques is more and more popular in the research community. In hybrid systems extraction often involves morphological normalization, so that each word (lexical item) can be identified regardless its actual morphological form. Two basic approaches to morphological normalization are stemming, where a word is transformed (usually heuristically) into its stem, and lemmatization, where a word is transformed into its base form by morphological analysis (Pecina, 2008).

In general, the process of multiword unit (NP including) extraction follows the steps of a) identification of candidates from the text and b) filtering the candidates. Particular hybrid extraction techniques differ in the amount and order in which linguistics and statistics are used.

(Smadja, 1993) creates a set of collocation candidates applying statistical co-occurrence information on a pretagged corpus and after extraction uses parsing for filtering out invalid results.

(Daille et al. 1994) make use of linguistic knowledge at the first stage of extraction to identify two-word noun phrases which correspond to a limited number of syntactic patterns on the previously tagged corpora. At the second stage statistical scores based on the number of occurrences of the pairs are used to select the "good" ones among the candidates.

(Seretan and Wehrli, 2006) use a syntactic parser in the first extraction stage for identifying two-word collocation candidates. The pairs are then partitioned according to their syntactic configuration. Finally, the log likelihood ratios test (Dunning, 1993) is applied to filter "good" NPs.

(Pecina, 2008) describes the extraction of two-word collocation candidates performed on morphologically normalized texts and filtered by a frequency filter and a part-of speech filter.

(Piao et al., 2005) suggest augmenting the power of multiword expression extraction by combining a statistical tool for searching and identifying English multiword expressions with a lexicon-based English semantic tagger (Rayson et al., 2004). The authors emphasize that training the tools on specific domains is essential for good results. Domain restrictions, such as strict structuring and sublanguage specificity is normally taken into consideration by various data and text mining tools applied to patent texts (Hull et al., 2001; Fattori et al., 2003).

[1] It is impossible to acquire knowledge including all words in all senses, a priori defined syntactic configurations and disambiguation rules.

[2] See, e.g., (Porter, 1980) for stemming algorithm.

3 Task definition

Our ultimate goal is to develop a methodology for extracting multiword NP terminology targeted to intelligent results and computationally attractive properties for facilitating and speeding up the development and maintenance of high-quality real-world MT systems.

The target of our extraction effort is thus defined by the intersection of five criteria: (i) multiword expression, (ii) noun, (iii) terminology, (iv) increase of recall and precision, (v) reduction of computational cost. For this work, we considered a string composed of several words to be a multiword expression if its meaning cannot be computed from its elements (Gross, 1986). However, in this definition, we, similar to (Laporte et al.,2008) consider that the possibility of computing the meaning of phrases from their elements is of any interest only if it is a better solution than storing the same phrases in lexicons.

We extracted only expressions belonging to the noun part of speech. We recognized them through the usual criteria regarding their morphosyntactic context.

We assumed that multiword noun expressions in a technical text are terms, see, e.g., (Daille,1994) for similar approach.

We aim to extract an NP candidate which is included into a larger NP candidate, only if the shorter NP functions individually in the processed corpus or text meant for MT. For example, if we have NP candidates such as,

1. antenna port selection method
2. antenna port selection
3. antenna selection method
4. port selection method
5. antenna port
6. selection method
7. port selection

then if candidates 2, 4, 6, 7 do not function individually in the corpus, only candidates 1, 3 and 5 will be included in the final output.

Such approach has obvious advantages in domain-tuned MT, e.g., for saving multilingual lexicon acquisition effort. This restriction, however, can always be lifted if necessary.

Our ambition is not to loose low frequency and unique NP terms.

We experimented with different proportions of statistical and linguistic knowledge in an attempt to increase recall and precision and reduce computational cost.

4 Approach

We tried and discarded the idea of starting extraction with stoplist words removal as it may lead to "bad" combinations of words. For example, removal of stoplist words (boldfaced) at the preprocessing stage from the patent fragment:

...**a** table **in which the** wireless location system continuously maintains **a** copy **of the** status **of** transmitters…

will lead to extraction of such "NPs" as

*table wireless location system
*copy status transmitters

Such combinations will not be filtered out automatically as they satisfy our grammar and they could have a high frequency due to the specificity of patent texts, where text fragments as above can be highly repetitive, e.g., in patent claims[3]. Even if unique they will not be discarded as our intention is to extract all NP terms.

We also decided against morphological normalization as preprocessing. Heuristic stemming algorithms, may fail to identify inflectional variants and lead to the extraction of wrongly combined and/or truncated character strings which are impossible to understand thus lowering precision. Proper NLP lemmatization is very expensive computationally. For these reasons we postponed lemmatization to the very last stage of processing.

We first calculate n-grams ($0<n<5$)[4] on a raw text, and then select NP candidates (singular and plural) with a strongly lexicalized constraint-based grammar as the major filtering mechanism. This initial candidate set is filtered by a count-based criterion. The key proposals here are:

- to apply shallow parsing based on grammar rules related to NP word order constraints to the raw text n-grams, and
- to apply these constraint rules through direct lexical (word string) match of an n-gram component against a lexicon rather than through POS tagging.

[3] A patent claim is a part of a patent where all essential features of an invention are formulated. A patent may include more than a hundred of claims.

[4] This is the most widely used limit for the number of words in n-gram extraction, but in our system "n" can be set to any number.

4.1 Resources

Our multiword NP extraction is based on the following resources:

- a shallow patent domain lexicon of part-of speech-unambiguous wordforms.
- application specific rules of a strongly lexicalized constraint grammar
- a heuristic noun lemmatizer

The shallow lexicon is a patent domain corpus-based list of wordforms with their part-of-speech information. The specificity of this lexicon is that it only includes part-of-speech-unambiguous wordforms. The advantage of using such a lexicon is in avoiding a computationally (and resource) expensive procedure of part-of-speech disambiguation. To build the lexicon we extracted part-of-speech-unambiguous lists of English wordforms from the lexicon of the patent MT system (Sheremetyeva, 2007).

Constraint-based grammar formalism is formalism in which a class of constraints is used to reduce a class of potential representations to the representations, which are well formed, or grammatical (Daniels and Meurers, 2004). In our approach we use the Phrase Structure Grammar constraints on the NP word order to select the initial set of NP candidates.

The specificity of our rules is that they are not the usual part-of-speech NP patterns to find n-grams which match these patterns. Our rules find those n-grams which cannot be NPs, without determining their full part-of-speech structures.

The rules are only applied to the first, last or middle (in case of 3- and 4-grams) words of an n-gram performing shallow (hence less expensive) n-gram parsing rather than complete parsing. The rules are as follows:

Rule 1
IF the first word in an n-gram is
Determiner/verb/preposition/wh-word/
THEN delete n-gram

Rule 2
IF last word in an n-gram is
adjective/verb/preposition/wh-word/article/
THEN delete n-gram

Rule 3.
IF the word, which is neither the first word,
nor the last word in a 3-gram,
is determiner/verb/wh-word
THEN delete the 3-gram

Rule 4
IF the word, which is neither the first word,
nor the last word in a 4-gram,
is /verb/wh-word
THEN delete the 4-gram

As can be seen, the rules do not exactly allow for all linguistically legal NP patterns. For example, the "determiner" constraint in Rule 1 is included because we do not want to extract NP candidates starting with articles or other determiners ("this", "that", etc.) as they are not included in MT lexicons.

Such application-motivated constraints are to some extent equivalent to stop words in traditional statistical approaches but in our case they are applied selectively and filter out inappropriate phrases rather than output non-existing NPs.

Due to the postponing of morphological normalization to the very last stage of processing, when an NP candidate set is supposed to consist of NPs only (plural and/or singular) we can afford to use a restricted noun lemmatizer rather than a full-fledged morphological lemmatizer. This again contributes a lot to processing robustness and resource/computation savings.

4.2 Procedure

The multiword NP terminology extraction starts by simple calculation of raw text n-grams, $0<n<5$[5]. Note, that though our goal is to extract multiword NPs we do not discard the list of 1-grams at this stage.

We then apply the rules of our grammar to filter out n-grams which cannot be NPs and build an initial set of NP candidates. The matching procedure in rule application is reversed. It starts with trying to match the first, last and middle word of every n-gram against the lexicon.

In case a lexical match is found the morphological description of the matching word is checked. If the matching word in the lexicon has a part-of-speech forbidden by the rules, the n-gram is discarded; otherwise it is added to a candidate set. If no lexicon match is found for any of the n-gram components the n-gram is also assigned an NP candidate status thus making the rules absolutely robust. Another advantage of the grammar rules is that they are computationally simple.

[5] This is the most widely used limit in n-gram calculation, but actually "n" can be set to any number.

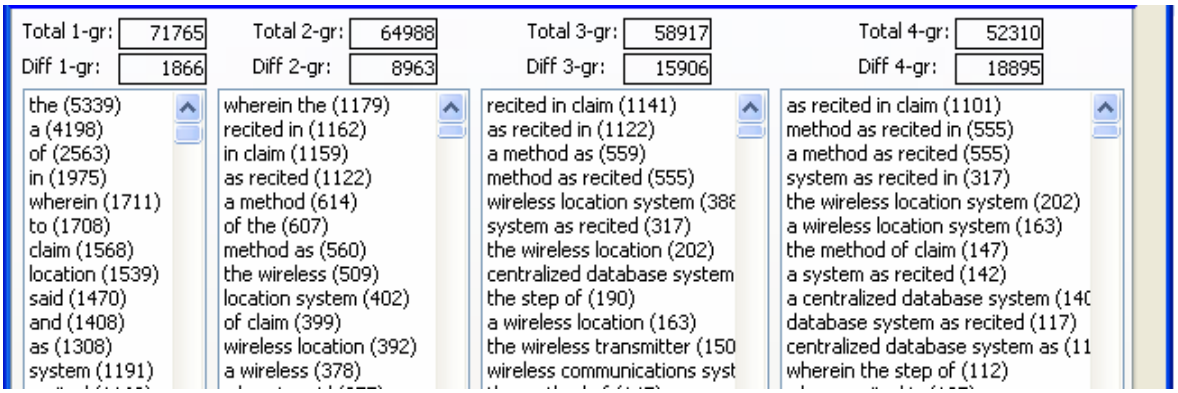


Figure 1. A fragment of top 1- to 4-gram lists before the application of the lexicalized constraint grammar rules. Numbers in brackets show frequencies.

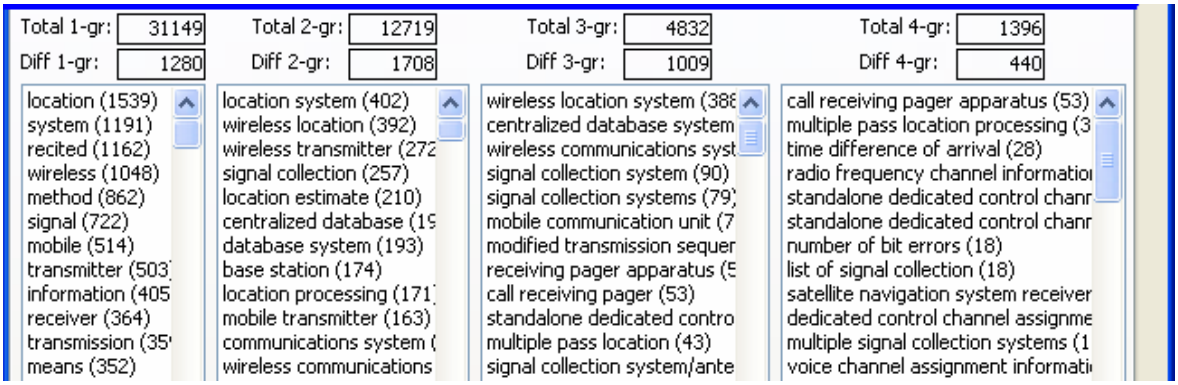


Figure 2. A fragment of top 1- to 4-gram lists after the application of the lexicalized constraint grammar rules. Numbers in brackets show frequencies.

Every rule taken separately will let pass some of the ill-formed NP candidates for which no match in the lexicon was found, as a lot of words being part-of-speech ambiguous are simply not in the application lexicon. However, successive application of the grammar rules to different words of the same n-gram compensates for this lack of the lexicon coverage. A "bad" NP not identified by one rule will be identified by another and thus discarded.

For example, the 3-gram "change the system" will not be forbidden by Rule 1, as the ambiguous word "change" (it can be a verb or a noun) is excluded from our lexicon, but this 3-gram will still be discarded by Rule 3, which demands to discard 3-grams containing determiner ("the" in this case) in the middle.

We thus can handle NP word order constraints in a computational parsing, without invoking additional layers of representation (i.e., disambiguated tagging).

The quality of filtering with our application modified lexicalized PHSG can be judged by comparing the n-gram lists in Figures 1 and 2, which show the top of 1- to 4-gram lists before and after the application of the grammar rules.

At the next stage of processing we create an expansion matrix over the initial set of all grammar filtered candidates, 1-gram including. A fragment of an expansion matrix is shown in Figure 3.

The matrix is created to make a decision whether shorter NP candidates which are parts of longer NPs function individually and "have the right" to be included in the final output.

For this purpose we introduce the count-based criterion "Uniqueness" (U) which is defined as the difference between an n-gram frequency and the sum of frequencies of its (n+1)-gram expansions.

A low U-value shows that the candidate is unlikely to be used individually. We experimentally selected the U=0 or U< 0 values as thresholds for filtering out undesired candidates.

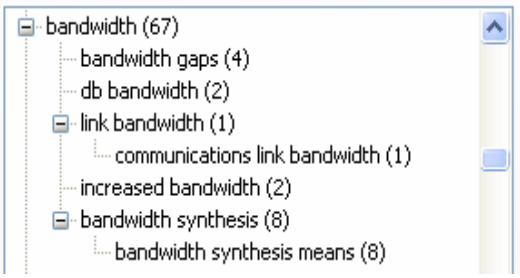


Figure 3. The expansion matrix of the 1-gram "bandwidth". Given in brackets are frequencies.

For example, in Figure 3 the candidates `link bandwidth` and `bandwidth synthesis` have U=0 and will be discarded; the rest 5 multiword candidates will be included in the candidate list for further filtering. After cleaning candidate duplicates in the expansion matrix [6] we once again run the grammar filter to discard the residue of "bad" candidates and then apply our noun lemmatizer. The duplicates[7] cleaned, the resulting set is output. Fragments of the output with top and low frequency NP terms are shown in Figures 4 and 5. Given in brackets are frequencies.

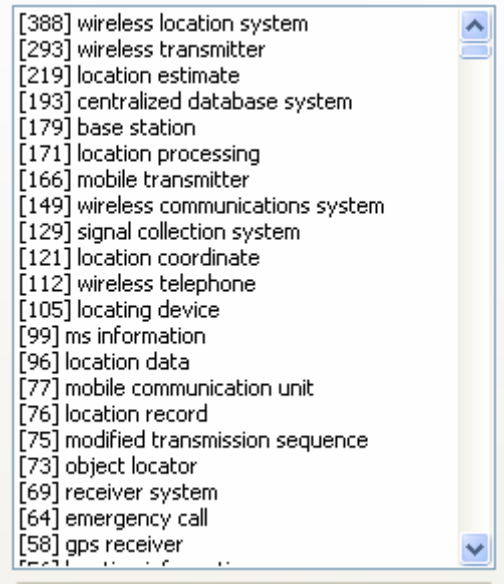


Figure 4. Top frequency multiword NP terms extracted over the evaluation corpus.

[6] One and the same candidate can appear as expansion in different n-gram nests.

[7] When a term in plural is lemmatized it may duplicate the term in singular which was already in the candidate set.

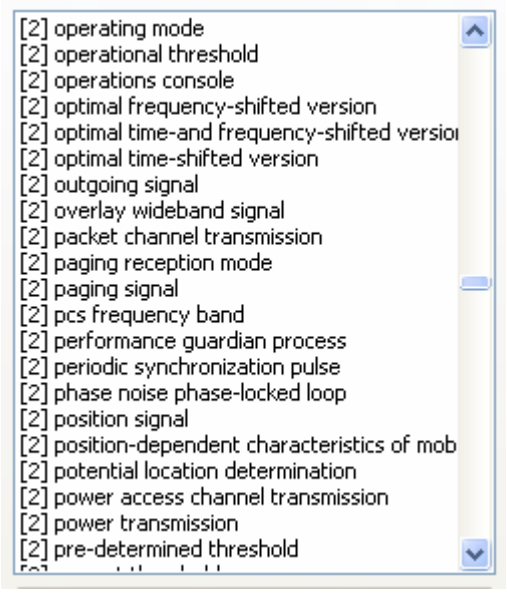


Figure 5. Low frequency multiword NP terms extracted over the evaluation corpus.

To summarize the extraction procedure is as follows:

1. IDENTIFICATION OF CANDIDATES
   a. Calculating raw text n-grams.
2. FILTERING
   a. Fist filtering of candidates with the use of the lexicon and constraint grammar rules.
   b. Calculation of an extension matrix
   c. Second filtering of candidates with the U-criterion
   d. Cleaning the resulting list (removing duplicates)
   e. Third filtering of candidates with the use of the lexicon and constraint grammar rules
   f. Lemmatization of resulting NPs
   g. Cleaning of the resulting list (removing duplicates)
3. OUTPUT

5 Evaluation

Our evaluation scheme covered the two basic demands: quality and speed. The quality evaluation method consisted in comparing our result list with a gold reference list.

The gold list was built manually by linguist students following the guidelines formulated in Section 3. The evaluation was performed over a patent corpus of 72000 words for which it was feasible to create a gold standard. The results of the evaluation are given in Table 1.

Total number of gold NPs	1425
Total extracted phrases	1476
Task correct NPs	1351
Gerundial phrases	43
Short NPs, not used individually	58
Missed NPs longer than 4 words	52
Missed NPs shorter than 4 words	16
Incorrect phrases	24

Table 1. Evaluation results.

Gerundial phrases are those like given below in bold face:

a server for **receiving tasking requests** from other applications

Such phrases if not actually NPs can still be attributed to nominal terminology as they normally mean processes and translationwise often correspond to regular NPs in other languages.

Most of short NPs, not used individually appeared in the final output due to "technical" reasons, namely, because we limited ourselves to a 4-gram window which does not allow for extracting NP terms containing more than 4 words. This makes it impossible to properly calculate the U value of shorter terms included in long ones. Examples of such terms are shown below (extracted NPs are in bold face).

multiple discrete frequency **elements of consistent amplitude**

outgoing **real time two-way communication**

caller **generated wireless local loop** communication system

One way to fix this problem is to widen the extraction window which might increase the computation time, but whether it really matters is left for further experiments. On the other hand, the shorter NPs, though not functioning individually, can still be included in an MT lexicon leaving translation of longer phrases to translation grammars. The number of such long terms is not very large, - 52 out of 1425 in our test.

The numbers of "bad" mistakes are shown in the last two rows on Table 1.

The speed of NP extraction is to a great extent increased due to the computational savings provided by our approach which removes a lot of n-grams from further computations at the early stage of extraction (compare numbers given on the top of Figures 1 and 2) and users shallow parsing and restricted lemmatizer.

In addition to that the extraction speed depends upon such factors as the load on the server, the speed of the network, and the size of the input text. Patents range in size from a few kilobytes to 1.5 megabytes. We can report that on a regular Hewlett-Packard X86-based PC it usually takes a fraction of a second to process a patent. An XML file of 8 megabytes containing 150 patents is processed in less that 2 min.

6 Conclusions

In this paper we described a methodology for extraction of multiword NP terms. The methodology provides for intelligent output and has computationally attractive properties due a specific combination of statistical, NLP and heuristic techniques. It includes n-gram calculation, shallow parsing based on strongly lexicalized constraint grammar. The grammar rules are applied to raw text n-gram components through direct lexical (word string) match against a non-ambiguous lexicon.

The methodology is robust as it does not depend on lexicon coverage and excludes such statistically or NLP expensive techniques as vast combinatorial computations or proper tagging and parsing.

We illustrated the approach on the example of patents in the English language but preliminary experiments show that it is portable to different domains and languages.

Different applications can benefit from the techniques proposed here, ranging from knowledge acquisition for RBMT systems or phrase-based SMT systems to machine-aided NLP tools.

We plan to extend this work in a number of ways. We are currently working on including the NP extractor into the analysis module of a patent MT system.

Another perspective is to extent the application to a multilingual keyphrase extraction tool for further use in multilingual search and information extraction.

References

Daille Béatrice, Éric Gaussier, & Jean-Marc Langé. 1994. Towards automatic extraction of monolingual and bilingual terminology. *Coling 1994: the 15th International Conference on Computational Linguistics*: Proceedings, August 5-9, 1994, Kyoto, Japan; pp. 515-521.

Daille Béatrice and Emmanuel Morin. 2008. A effective compositional model for lexical alignment. *IJCNLP 2008: Third International Joint Conference on Natural Language Processing*, January 7-12, 2008, Hyderabad, India; pp 95-102.

Daniels, M. and Meurers, D. 2004. GIDLP: A grammar format for linearization-based HPSG. HPSG04 Conference proceedings.

Dunning Ted . 1993. Accurate methods for the statistics of surprise and coincidence. Computational Linguistics, 19(1):61–74

Fattori, Michele., Pedrazzi Giorgio., Turra, Roberta. 2003. Text mining applied to patent mapping: a practical business case. World Patent Information,25

Gross, M. 1986. Lexicon-Grammar. The representation of compound words. In Proceedings of the 11th International Conference on Computational Linguistics,COLING'86, Bonn, West Germany, pp. 1-6.

Hewavitharana Sanjika, Alon Lavie, and Stephan Vogel. 2007. Experiments with a noun-phrase driven statistical machine translation system. *MT Summit XI*, 10-14 September 2007, Copenhagen, Denmark. *Proceedings*; pp.247-253.

Hull D., A i it-Mokhtar, S., Chuat, M., Eisele, A,, Gaussier, E., Grefenstette, G. 2001. Language technologies and patent search and classification. World Patent Inf 23.

Koehn P., F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In Proc. of the 41th Annual Meeting of the Association for Computational Linguistics.

Laporte, Eric, Takuya Nakamura and Stavroula Voyatzi. 2008. A French Corpus Annotated for Multiword Nouns. Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008) pp.27-30.

Pecan, Pavel. 2005. An extensive empirical study of collocation extraction methods. In Proceedings of the ACL Student Research Workshop, pp 13–18.

Pecina, Pavel .2008 Reference Data for Czech Collocation Extraction. Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008) pp.11-14..

Piao, S. L., Rayson, P., Archer, D. and McEnery, T. 2005. Comparing and Combining A Semantic Tagger and A Statistical Tool for MWE Extraction. *Computer Speech & Language* Volume 19, Issue 4, pp. 378-397

Pohl, Gábor. 2006. English-Hungarian NP alignment in MetaMorpho TM. *EAMT-2006: 11th Annual Conference of the European Association for Machine Translation*, June 19-20, 2006, Oslo, Norway. Proceedings; p.69-74

Rayson, P., Archer, D., Piao, S., and McEnery, T., 2004. The UCREL semantic analysis system. In: Proceedings of the LREC-04 Workshop, beyond Named Entity Recognition Semantic Labelling for NLP Tasks, Lisbon, Portugal, pp.7–12.

Seretan, Violeta and Eric Wehrli. 2006. Accurate collocation extraction using a multilingual parser. *Coling-ACL 2006: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, 17-21 July 2006; pp.953-960.

Sheremetyeva, Svetlana. 2007. On Portability of Resources for Quick Ramp-Up of Multilingual MT for Patent Claims. Proceedings of the workshop on Patent Translation in conjunction with MT Summit XI, Copenhagen, Denmark, September 10-14.

Smadja, F. 1993. Retrieving collocations from text. Xtract. *Computational Linguistics* 7(4):143–177.

Tomokiyo, T., and Hurst, M. 2003. A language model approach to keyphrase extraction. Proceedings of ACL Workshop on Multiword Expressions.

Zens, R., Och, F.J., Ney, H.: Phrase-based statistical of LNCS. Springer Verlag (September 2002) machine translation. In: Advances in artificial intelligence. 25. Annual German Conference on AI. Volume 2479 18–32.

Rule-Based Augmentation of Training Data in Breton–French Statistical Machine Translation

Francis M. Tyers

Dept. de Llenguatges i Sist. Informàtics,
Universitat d'Alacant
E-03071 Alacant (Spain)

Prompsit Language Engineering
Av. St. Francesc d'Assís, 74, 1r-L
E-03195 l'Altet (Spain)

ftyers@prompsit.com

Abstract

This article describes an initial statistical machine translation system between Breton, a Celtic language spoken in France, and French. It also describes a method for leveraging existing resources from an incomplete rule-based machine translation system to improve the coverage and translation quality of the statistical system by generating expanded bilingual vocabulary lists. Results are presented which show that the use of this method improves the results of the system with respect to both the baseline, and the baseline with a lemma-to-lemma bilingual lexicon.

1 Introduction

Breton is a Celtic language of the Brythonic branch largely spoken in Brittany in the north-west of France. Historically it was spoken only in the northern part of Brittany, *Breizh-Izel* (Lower Brittany). This contrasts with *Breizh-Uhel* (Higher Brittany) which is traditionally Romance-speaking.

Although some sources put the number of native speakers at between 500,000 and 600,000 (Gordon, 2005), a more up-to-date estimate can be found from the organisation *Ya d'ar brezhoneg* which gives a number of 201,083 as of the 11th November 2008, and states that the number is decreasing at a rate of at least one per hour.[1] Breton is classed as a language in serious danger of extinction by the *UNESCO Red Book on Endangered Languages* (Salminen, 1999), a situation exacerbated by the *laissez-faire* policies of the French state.

Like other Celtic languages, Breton exhibits the phenomenon of initial consonant mutation. This occurs when the initial consonant of a word changes based on morpho-syntactic context. For example in the word *tad* "father", the initial consonant mutates to a 'z' (*aspirant* mutation) when the word follows the possessive *ma* "my", so *tad* is "father", while "my father" is *ma zad*.

As for many less-resourced language pairs, while there is little aligned bilingual text, bilingual lexicons are more readily available. One solution would be to use these bilingual lexicons within a rule-based system that makes use of the features found in the bilingual lexicon: part-of-speech, gender, number etc. to try to compensate for the lack of data with some level of generalisation. Even if little parallel data is available, it is still worthwhile to compare any attempt at a more linguistically motivated system with a greater generalising power with a straight-forward, state-of-the-art non-linguistic approach, such as phrase-based statistical machine translation (SMT).

2 Resources

2.1 Parallel corpora

For any language pair, parallel corpora are the scarcest of all resources. In the case of a language with a small population of speakers and no official recognition, the hurdle is even greater. In contrast with Welsh, there are no bilingual parliamentary proceedings that may be used. As an official body for the defence of the Breton language, the *Ofis ar Brezhoneg* is a big producer of Breton translations and we were given the opportunity to access their translation memories, which

[1] http://www.yadarbrezhoneg.com/?article245; Accessed: 11th November, 2008

Proceedings of the 13th Annual Conference of the EAMT, pages 213–217,
Barcelona, May 2009

Corpus	Number of aligned segments
training	27,987
tuning	1,000
devtest	1,000
test	1,000

Table 1: Split of parallel corpus

mostly contain short segments (with an average length of approx. 9 words per segment) largely in the domains of tourism and computer localisation. This results in approx. 285,000 Breton words and 282,073 French words distributed in 31,000 lines. After basic space- and punctuation-based tokenisation, the total number of distinct tokens for Breton was 36,435 and for French was 41,932. These were split into training, tuning and two test sets as described in table 1.[2]

3 A rule-based system

A rule-based MT system for Breton–French is currently being developed inside the Apertium project.[3] Apertium (Armentano-Oller et al., 2006) is an open-source platform for creating rule-based machine translation systems. It was initially designed for closely-related languages, but has also been applied to work with more distant language pairs, such as Welsh–English (Tyers and Donnelly, 2009) and Basque–Spanish. The translation engine in the platform follows a largely shallow-transfer approach. Finite-state transducers are used for lexical processing, first-order Hidden Markov Models (HMMs) and optionally, Constraint Grammars (CGs) based on VISLCG3[4] are used for part-of-speech disambiguation, and multi-stage finite-state based chunking is used for structural transfer.

The current status of the Breton–French system is as follows: the system has a morphological analyser for Breton with approximately 11,000 lemmata (approx. 85% coverage on open-domain text), a bilingual dictionary with 10,797 part-of-speech tagged correspondences between Breton and French, and a very small number of transfer rules (e.g. for concordance and re-ordering within noun phrases, verbal conjugation and pronoun insertion) adapted from the Spanish–French language pair. It is not currently considered a production system as the coverage of the transfer rules is very sparse.

For examples of entries from the morphological analyser and bilingual lexicon, please see figures 1 and 2 respectively. In the morphological analyser, there are two kinds of paradigms (`<par>`) referenced, the first for specifying the initial consonant mutations described above, the second for listing all the morphological forms of a given word along with their analyses. For example, in the case of verbs, a single combination of lemma and paradigm generates between 37 surface forms (for unmutating initial consonants) and 193 (for mutating initial consonants).

4 A statistical phrase-based system

A phrase-based statistical model was trained using the training and tuning sets mentioned above. Although other language model software is frequently used in the literature, the IRSTLM (Marcello et al., 2008) implementation was chosen as it was available and open-source. A 3-gram language model was trained using the French side of the parallel data. The rest of the training process followed the instructions for the baseline system for WMT08, the shared task in the ACL 2008 workshop on statistical machine translation (Callison-Burch et al., 2008). Only a few modifications in the tokeniser provided were necessary, to deal with the *c'h* character in Breton. The training and tuning corpora were tokenised and lowercased to try to alleviate the data sparseness. BLEU scores optimised with the MERT algorithm (Och, 2003) on the tuning set and obtained on the test set are displayed in table 2.

5 Extending the parallel corpus

As the corpus used for training was much smaller than usually used in SMT, there was a problem of coverage. This was aggravated by the fact that Breton is an inflected language and as mentioned previously also exhibits the phenomenon of initial consonant mutation. Such a small corpus is unlikely to contain the majority of frequent surface forms, and almost certainly would not contain the less frequent ones.

To try and alleviate the problem of low coverage of the training data, it was decided to make use of the resources available in the nascent rule-based system described above. Two approaches

[2]The data used in the experiment may be downloaded from `http://elx.dlsi.ua.es/~fran/brfr_OAB_corpus.tgz` and used under the terms of the GNU GPL.
[3]`http://www.apertium.org/`
[4]`http://visl.sdu.dk/constraint_grammar.html`

```
<e lm="labourat">
  <i>labour</i>
  <par n="labour/at__vblex"/>
</e>
<e lm="kadarnaat">
  <par n="initial-k"/>
  <i>adarna</i>
  <par n="labour/at__vblex"/>
</e>
```

Figure 1: Example of morphological analyser entries for two verbs (*labourat* 'to work' and *kadarnaat* 'to confirm'), including inflectional paradigm (`labour/at__vblex`) and mutation paradigm (`initial-k`)

```
<e>
  <p>
    <l>labourat<s n="vblex"/></l>
    <r>travailler<s n="vblex"/></r>
  </p>
</e>
<e>
  <p>
    <l>kadarnaat<s n="vblex"/></l>
    <r>confirmer<s n="vblex"/></r>
  </p>
</e>
```

Figure 2: Example of bilingual lexicon entries for two verbs. The bilingual lexicon specifies correspondences between lemmata and parts of speech.

were taken. The first was to simply add the bilingual transfer lexicon from the system to the end of the training data. This consisted of 10,797 lemmata. The second was to automatically generate appropriate mappings between all of the surface forms of the given lemmata in the dictionaries of this system.

There has been existing research in this area, for example Dugast et al. (2008) generated a parallel corpus from a rule-based system to train a phrase-based system, and Schwenk (2009) uses an inflected dictionary to produce training data for a statistical system, albeit in a well resourced language pair (French–English).

In order to generate the surface-form mappings, an expansion of all possible surface forms was taken, along with analyses in the Breton morphological analyser. These analyses were then passed through the rest of the Apertium pipeline in order to produce all of the possible translations of surface forms in French. This produces a bilingual inflected dictionary (see figure 3). It is worth mentioning that as a result of the transfer rules, entries for verbs, are generated, where appropriate (e.g. finite verb tenses) with the corresponding subject pronoun in French, and Breton tenses which are not found in French are converted into French tenses (e.g. past habitual is converted to imperfect).

```
mignon,ami
mignoned,amis
vignon,ami
vignoned,amis
dale,retarde
dale,il retarde
labouren,je travaillais
...
```

Figure 3: Example of output from the dictionary expansion and translation – *mignon* 'friend', *dale* 'late' and 'He is delaying' and *labouren* 'I worked'

This 'expanded' bilingual dictionary of surface forms was added to the end of the training corpus, and consisted of 116,514 mappings of inflected Breton forms to inflected French forms.

6 Evaluation and error analysis

As time has not yet been found for a manual evaluation, below are presented the BLEU (Papineni et al., 2002) scores for the three statistical models described above, along with a baseline word-for-word translation generated by the unfinished RBMT system. As expected, the number of unknown words decreases when the bilingual lexicon is added to the training data, and even more when the fully-expanded bilingual lexicon is added. The rise in BLEU (keeping in mind that these are short sentences of ten words on average) is probably also due to side effects such as a better word alignment and a better French context available to the language model scoring. When comparing systems 3 and 4, a quick manual review may attribute most changes to plural forms.

See examples in table 3. The first example shows how the plural form of the Breton for *syllabe* (syllable) could be matched thanks to the morphological extension of the lexicon. In example 2, another kind of extension could be used by the decoder. In French, inflected verbs require the presence of subject pronouns, whereas in Breton this is not the case. This may lead to alignment errors

System	Description	BLEU	Phrase pairs	Unknown words in `devtest`
system 1	word-for-word	0.16	n/a	1,191
system 2	baseline phrase-based SMT	0.29	800k	623
system 3	+ uninflected dictionary	0.30	807k	562
system 4	+ inflected dictionary	0.36	843k	531

Table 2: BLEU scores

Example 1	*Benveg troc'hañ dre silabennoù*
ref	outil de césure par syllabe
gloss	hyphenation tool
system 3	outil coupe par silabennoù
system 4	outil de coupure par syllabes
Example 2	*E rankit kevreañ ouzh an holl darzhioù roadennoù*
ref	Vous devez vous connecter à toutes les sources de données
gloss	You should connect to all of the data sources
system 3	Devez connexion de données . les darzhioù
system 4	Vous devez se connecter tous les darzhioù de données
Example 3	*Emirelezhioù Arab Unanet*
ref	émirats arabes unis
gloss	United Arab Emirates
system 3	émirats arabes unies
system 4	émirats arabes unissez

Table 3: Translation examples

especially with sparse data. In this example, the second plural form in present tense of the Breton verb *rankout* (to have to), *rankit* was mapped to its French equivalent with the corresponding pronoun *vous devez*.

It is also worth noting that the error *se connecter* for *vous connecter* could be alleviated with a more robust verb generation. In French the verb is reflexive, and this is marked in the bilingual lexicon, but the appropriate reflexive pronoun is not yet generated by the rule.

In example 3, the translation of *Emirelezhio Arab Unanet* (United Arab Emirates) displays the adjective for "united" with the incorrect gender. System 4 does not perform better, since it insteads outputs the imperative form of the corresponding verb "unite!". It is very likely that in a real parallel corpora the correct translation (as an adjective) would have been more frequent than the one picked up here in decoding from the extended bilingual lexicon.

7 Conclusions and future work

This paper has presented, to my knowledge, the very first results on Breton to French machine translation. While comparing BLEU scores on a rule-based and a statistical system is not meaningful (Callison-Burch et al., 2006; Labaka et al., 2007), it has shown that the work on the linguistic coding of dictionary entries helped improve a statistical model that had to be trained on little data.

One of the avenues for improving the baseline statistical system would be to add a larger language model on the target side. It would probably also be possible to try to learn probabilities for the rule-based created phrase pairs as in Koehn and Knight (2000). Another option would be to try and create "expanded" phrases based on chunks extracted from a bilingual corpus. For example if you have *war toenn an ti*, "sur le toit de la maison" (on the roof of the house), it would be fairly straightforward given the rule-based system to generate all possible morphological combinations, viz. *war toennoù an ti*, "sur les toits de la maison" (on the roofs of the house), *war toennoù an tiez*, "sur les toits des maisons" (on the roofs of the houses), and *war toenn an tiez*, "sur le toit des maisons" (on the roof of the houses) respectively.

It is also worth noting that at present the Breton–French lexicon in Apertium has only one (gener-

ally the most frequent) translation per word. It would be feasible to generate more than one entry per word, and then score these on language models.

The method described here is knowledge-light, requiring only a morphological analyser, bilingual dictionary and some very basic transfer rules (for verb conjugation) and could be applied to other under-resourced language pairs to improve the coverage of a statistical system where little parallel data is available.

Acknowledgements

I am very grateful to the *Ofis ar Brezhoneg* for making available their translation memory, and for their consistent help during the project. I would also like to extend special thanks to: Fulup Jakez, the director, for his work on verifying and expanding the Breton morphological analyser and Breton–French lexicon, the two contributors to this paper who do not wish to be named, and the reviewers for the helpful comments I received.

References

Armentano-Oller, Carme. Carrasco, Rafael C., Corbí-Bello, Antonio M., Forcada, Mikel L., Ginestí-Rosell, Mireia, Ortiz-Rojas, Sergio, Pérez-Ortiz, Juan Antonio, Ramírez-Sánchez, Gema, Sánchez-Martínez, Felipe and Scalco, Miriam A. 2006. "Open-source Portuguese-Spanish machine translation" *Proceedings of the 7th International Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR-2006*

Callison-Burch, Chris, Osbourne, Miles and Koehn Philip 2006. "Re-evaluating the role of Bleu in machine translation research" in *11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 249–256

Callison-Burch, Chris, Fordyce, Cameron, Koehn, Philipp, Monz, Christof and Schroeder, Josh 2008. "Further Meta-Evaluation of Machine Translation" in *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 70–106

Dugast, Loïc, Senellart, Jean and Koehn, Philipp 2008. "Can we relearn an RBMT system?" in *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 175–178

Gordon, Raymond G., Jr. (ed.) 2005. *Ethnologue: Languages of the World, Fifteenth edition* (Dallas, Tex.: SIL International)

Koehn, Philip and Knight, Kevin 2000. "Estimating Word Translation Probabilities from Unrelated Monolingual Corpora Using the EM Algorithm" in *Proceedings of the Seventeenth National Conference on Artificial Intelligence* pp. 711–715

Koehn, Philip, Hoang, Hieu, Birch, Alexandra, Callison-Burch, Chris, Federico, Marcello, Bertoldi, Nicola, Cowan, Brooke, Shen, Wade, Moran, Christine, Zens, Richard, Dyer, Chris, Bojar, Ondrej Constantin, Alexandra and Herbst, Evan 2007. "Moses: Open source toolkit for statistical machine translation" in *ACL 2007, demonstration session*.

Labaka, Gorka, Stroppa, Nicholas, Way, Andy and Sarasola. Kepa 2007. "Comparing rule-based and data-driven approaches to Spanish-to-Basque machine translation" in *Machine Translation Summit XI*, Copenhagen, Denmark, pp. 297–304

Federico, Marcello, Bertoldi, Nicola and Cettolo, Mauro 2008. "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models" *Proceedings of the Interspeech 2008*, pp. 1618–1621

Och, Franz J. 2003. "Minimum error rate training in statistical machine translation" *41st Annual Meeting of the Association for Computational Linguistics* pp. 160–167

Papineni, Kishore, Roukos, Salim, Ward, Todd and Zhu, Wei-jing 2002. "BLEU: a method for automatic evaluation of machine translation" in *40th Annual meeting of the Association for Computational Linguistics* pp. 311–318

Salminen, Tapani 1999. *Unesco Red Book on Endangered Languages*

Schwenk, Holger 2009. "On the use of comparable corpora to improve SMT performance" to appear *EACL-2009*

Tyers, Francis M. and Donnelly, Kevin 2009. "apertium-cy: a collaboratively-developed free RBMT system for Welsh to English" *Prague Bulletin of Mathematical Linguistics* No. 91, pp. 57–66.

Can Semantic Role Labeling Improve SMT?

Dekai WU[1] **Pascale FUNG**[2]

Human Language Technology Center
HKUST
[1]Department of Computer Science and Engineering
[2]Department of Electronic and Computer Engineering
University of Science and Technology, Clear Water Bay, Hong Kong
dekai@cs.ust.hk pascale@ee.ust.hk

Abstract

We present a series of empirical studies aimed at illuminating more precisely the likely contribution of semantic roles in improving statistical machine translation accuracy. The experiments reported study several aspects key to success: (1) the frequencies of types of SMT errors where semantic parsing and role labeling could help, and (2) if and where semantic roles offer more accurate guidance to SMT than merely syntactic annotation, and (3) the potential quantitative impact of realistic semantic role guidance to SMT systems, in terms of BLEU and METEOR scores.

1 Introduction

In this investigative paper, we present a new set of empirical studies aimed at illuminating more precisely the likely contribution of semantic parsing and role labeling toward improving statistical machine translation accuracy.

The most glaring errors made by statistical machine translation systems continue to be those resulting in confusion of semantic roles. These sorts of translation errors often result in serious misunderstandings of the essential meaning of the source utterances — who did what to whom, for whom or what, how, where, when, and why.

It has been widely observed that the negative impacts of such errors on the utility of the translation are inadequately reflected by evaluation metrics based on lexical criteria. The accuracy of translation lexical choice has reached increasingly satisfactory levels—at least for largely literal genres such as newswire — which helps boost lexically oriented scores such as BLEU (Papineni *et al.*, 2002) or METEOR (Banerjee and Lavie, 2005) despite serious role confusion errors in the translations.

It has also often been noted that precision-oriented metrics such as BLEU tend to reward fluency more than adequacy (in particular, BLEU's length penalty is only an indirect and weak indicator of adequacy). Today's SMT systems produce translations that often contain significant role confusion errors but nevertheless read quite fluently.

Thus, while recent years have seen continued improvement in the accuracy of statistical machine translation systems as measured by such lexically based metrics, this underestimates the effect of the persistent errors of role confusion upon the actual translation utility.

This situation leads us to consider the potential application of shallow semantic parsing and semantic role labeling models to SMT, in ways that might reduce role confusion errors in the translation output. Within the lexical semantics community, increasingly sophisticated models for shallow semantic parsing are being developed. Such semantic parsers, which automatically label the predicates and arguments (roles) of the various semantic frames in a sentence, could automatically identify inconsistent semantic frame and role mappings between the input source sentences and their output translations. This approach is supported by the results of Fung *et al.* (2006), which reported that (for the Chinese-English language pair) approximately 84% of semantic role mappings remained consistent cross-lingually across sentence translations.

We approach this promise with caution, however, given the painful lessons learned through the historical difficulty of making syntactic and semantic models contribute to improving SMT accuracy. The past decade has at last seen increasing amounts of evidence that SMT accuracy can indeed be improved via tree-structured and syntactic models (e.g., Wu (1997); Chiang and Wu (2008); Wu and Chiang (2009)) despite numerous disappoint-

Proceedings of the 13th Annual Conference of the EAMT, pages 218–225,
Barcelona, May 2009

ing attempts Och *et al.* (2004). More recently, lexical semantics models for word sense disambiguation have also finally been successfully applied to increasing SMT accuracy (e.g., Carpuat and Wu (2007), Chan *et al.* (2007); Giménez and Màrquez (2007a)) again after surprising initial failures (e.g., Carpuat and Wu (2005)). In both the syntactic and semantic cases, improving SMT accuracy ultimately required making major adaptations to the original linguistic models. We can reasonably expect it to be at least as difficult to successfully adapt the even more complex types of lexical semantics modeling from semantic parsing and role labeling.

Avoiding the many potential blind alleys calls for careful analysis and evaluation of (1) the frequencies of types of SMT errors where semantic parsing and role labeling could help, (2) if and when semantic roles offer more accurate guidance to SMT than merely syntactic annotation, and (3) the potential quantitative impact of realistic semantic role guidance to SMT systems, at least in terms of scores such as BLEU and METEOR.

In this paper, we present a series of four experiments designed to address each of these questions, using Chinese-English parallel resources, a typical representative SMT system based on Moses, and shallow semantic parsers for both English and Chinese.

2 Related work

While this is a new avenue of inquiry, the background relevant to the experiments described here includes (1) a broad body of work on shallow semantic parsing and semantic role labeling, the majority of which has been performed on English, (2) a relatively small body of work specific to semantic parsing and semantic role labeling of Chinese, and (3) a proposal to measure semantic role overlap as one of the key factors in new MT evaluation metrics.

2.1 Shallow semantic parsing

Semantic parsers analyze a sentence with the aim of identifying the "who did what to whom, for whom or what, how, where, when, and why." Shallow semantic parsing extracts the predicate-argument structure of verbs in a sentence based on the syntactic tree of that sentence. For example, the predicate argument structure of the verb *hold* in Figure 1 specifies a "holding" relation between *both sides* (who) and *meeting* (what) *on Sunday* (when). For a sentence with multiple verbs, there can be multiple predicate argument structures.

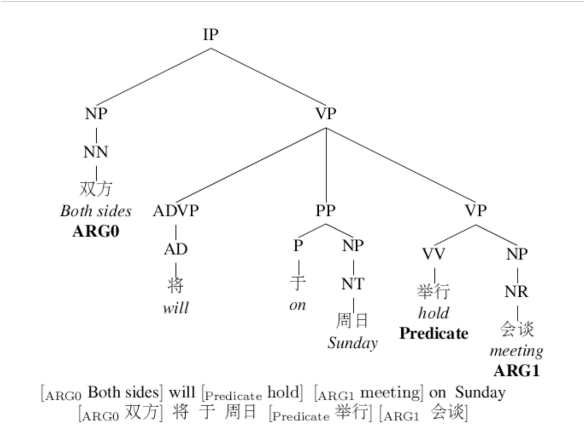


Figure 1: Chinese shallow semantic parsing example.

Shallow semantic parsing systems are mostly based on classifiers that learn from a manually annotated semantic corpus (Gildea and Jurafsky (2002), Pradhan *et al.* (2005)). Following the publication of the Proposional Bank (PropBank) (Palmer *et al.*, 2005) first in English, then in Chinese, it has been possible to train these classifiers to perform semantic analysis on news wire type of texts.

2.2 Chinese shallow semantic parsing

Systems that perform shallow semantic parsing on Chinese texts are likewise based on classifiers and trained on the Chinese PropBank and the bilingual Chinese-English Parallel PropBank (Sun and Jurafsky (2004), Xue (2006), Fung *et al.* (2006)). It is interesting to note that, despite the very different characteristics of Chinese verbs (Xue and Palmer, 2005) from those in English, the core algorithm of a shallow semantic parser remains the same. As was found to be the case in English, SVM classifiers have been found to outperform maximum entropy classifiers for this task (Fung *et al.*, 2006). The primary difference lies in the feature set chosen to represent semantic information.

In experiments carried out on PropBank data using gold standard syntactic parse trees, extended syntactic features such as Path Trigram and Path Abbreviations were found to have the highest contribution to system performance (Fung *et al.*, 2006). Another feature, Verb Cluster, was also found to be most useful by Xue and Palmer (2005).

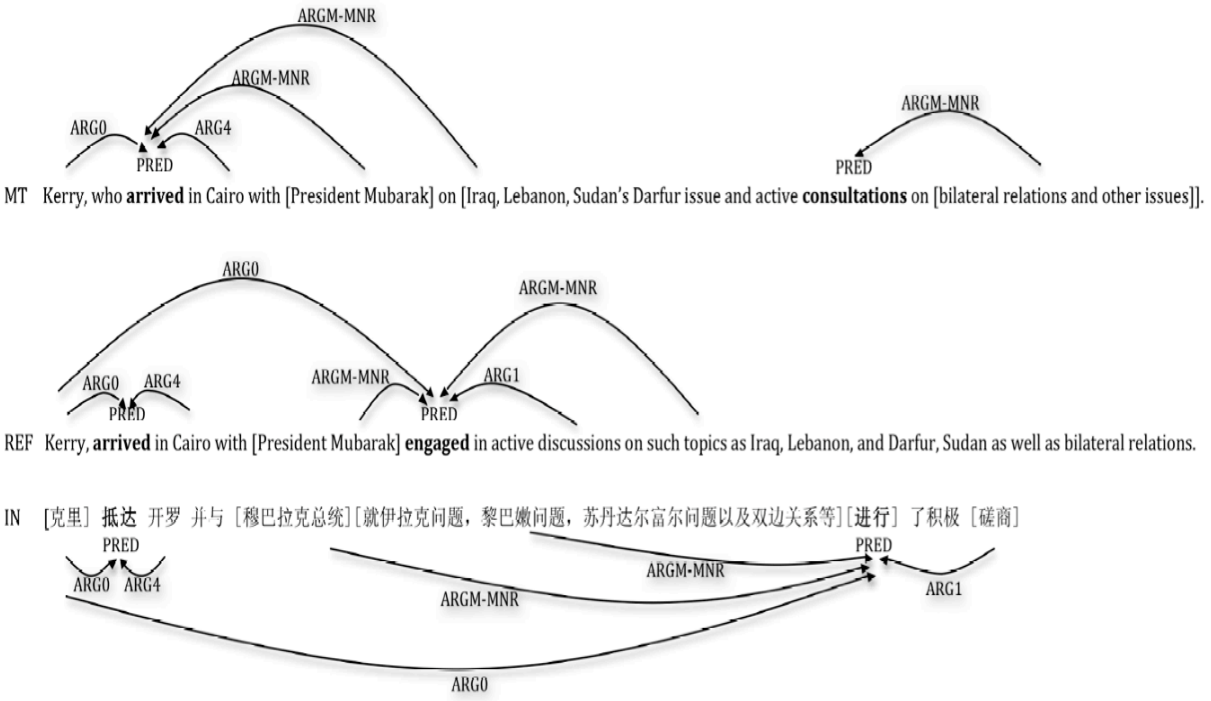


Figure 2: Example of semantic frames in Chinese input and English MT output.

2.3 MT evaluation metrics based on semantic role overlap

Giménez and Màrquez (2007b) and Giménez and Màrquez (2008) introduced and refined a set of new MT evaluation metrics employing rich assortments of features reflecting various kinds of similarity at lexical, shallow syntactic, deep syntactic, shallow semantic, and deep semantic levels.

Under a number of scenarios—particularly the out-of-domain scenarios—measuring the overlap of shallow semantic roles between the source and target language sentence pairs contributes to improved correlation with human judgment of translation quality. Unsurprisingly, measuring the overlap of manually annotated deep semantic relations contributes even more in some scenarios. However, given the state of automatic semantic parsing technology, realistically we are today still much closer to being able to incorporate automatic shallow semantic parsing into working SMT systems, and thus we focus on shallow semantic parsing and semantic role labeling for the present.

3 Semantic frames in SMT output

The first of the experiments aims to provide a more concrete understanding of one of the key questions as to the role of semantic parsing in SMT: how well do typical current SMT systems already perform on semantic frames?

The annotated example in Figure 2 shows, from bottom to top, (IN) a fragment of a typical Chinese input source sentence that is drawn from newswire text, (REF) the corresponding fragment from its English reference sentence, and (MT) the corresponding fragment of the output sentence from a state-of-the-art SMT system.

A relevant subset of the semantic roles and predicates has been annotated in these fragments. In the Chinese input and its corresponding English reference, there are two main verbs marked PRED. The first, (*arrived*), has two arguments: one in an ARG0 agent role, (*Kerry*); and another in an ARG4 destination role, (*Cairo*). The second verb, (*engaged*), has four arguments: one in an ARG0 agent role, again (*Kerry*); one in an ARG1 role, (*discussions*); and two others in ARGM-MNR manner roles, (*with Mubarak*) and (*on topics*).[1]

In contrast, in the SMT translation output, a very different set of predicates and arguments is seen. While the PRED *arrived* still has the same correct ARG0 *Kerry* and ARG4 *Cairo*, now the ARGM-MNR manner role with *President Mubarak* is incorrectly modifying the *arrived*, instead of an engaged predicate. In fact, the *en-*

[1]Minor variations on the role labeling in these examples are possible, but not central to the present point.

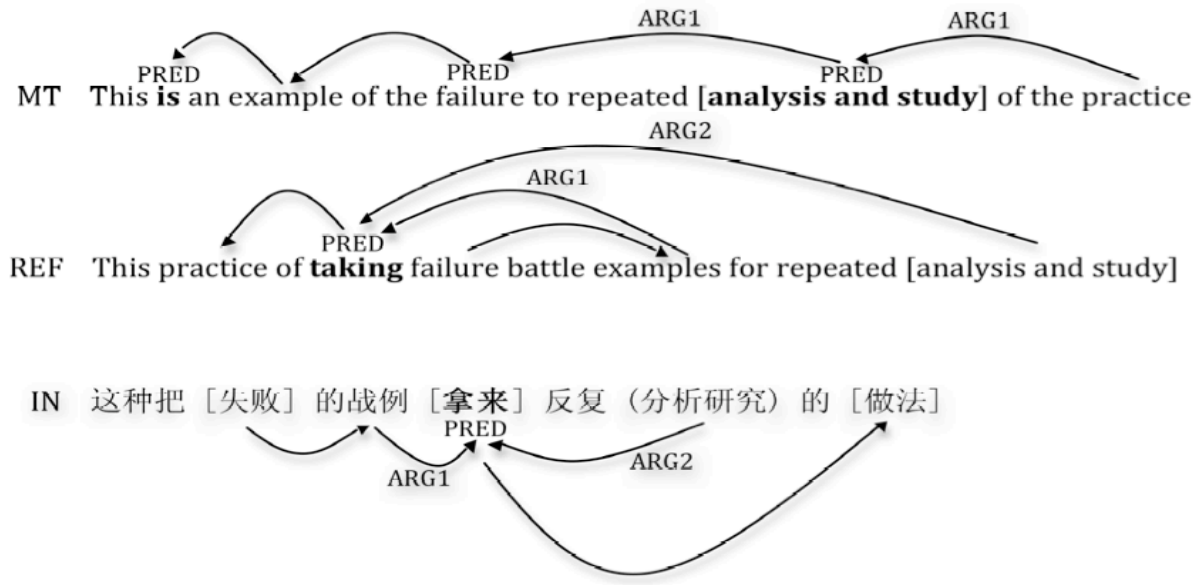


Figure 3: Example of semantic frames in Chinese input and English MT output.

gaged predicate has erroneously been completely dropped by the SMT system, so there is no verb to which the arguments of *engaged* can be attached.

Figure 3 shows another typical example. Again, PRED marks the main verb in the Chinese input source fragment and its corresponding English reference, (*taking*). It has two arguments: an ARG1 (*battle examples*) and an ARG2 (*analysis and study*).

The SMT translation output, however, not only lacks the main verb, but includes many incorrect predicates and roles. Such spurious predicate-argument structures are clearly seriously detrimental to even cooperative readers straining to guess the meaning of the original Chinese.

3.1 Experimental setup

To assess the above sorts of phenomena quantitatively, we designed an experiment making use of 745 bi-sentences extracted from the Parallel PropBank with gold standard annotations of both syntactic and semantic roles.

We use the Chinese sentences as system input and their corresponding English translations as the reference translations. We use the open source statistical machine translation decoder Moses (**?**) for the experiments, translating the PropBank Chinese sentences into English with the same model trained for our participation in the IWSLT 2007 evaluation campaign (Shen *et al.*, 2007).The English translations generated by the decoder are the system output. Based on the system input and the reference translation, we intend to investigate whether the predicate verbs are correctly translated and their predicate-argument structures preserved in the system output.

Table 1: Accuracy of predicate-argument structure in Chinese-English SMT output for data set A.

P-A Structure	Precision	Recall	F-measure
Predicate	0.98	0.57	0.72
ARG0	0.74	0.38	0.50
ARG1	0.73	0.41	0.53
ARG2	0.82	0.32	0.46
ARG3	1.00	0.67	0.80
ARG4	1.00	0.33	0.51
All ARGs	0.74	0.39	0.51

We first randomly select 50 bi-sentences, without any constraint on the translation accuracy of the predicate verbs, to form the first observation data set (data set A).

3.2 Experimental results

Human evaluation of these results show that, for all 138 predicate verbs in the system input (Chinese sentences), only 79 (around 57%) of them are correctly translated in the system output; and given such correctly translated predicate verbs, the translation of their semantic arguments can only achieve around 51% overall F-measure. The detailed results are shown in Table 1.

Table 2: Accuracy of predicate-argument structure in Chinese-English SMT output for data set B.

P-A Structure	Precision	Recall	F-measure
Predicate	1.00	1.00	1.00
ARG0	0.83	0.66	0.74
ARG1	0.84	0.78	0.81
ARG2	0.80	0.78	0.38
ARG3	0.00	0.00	N/A
ARG4	0.50	1.00	0.66
All ARGs	0.84	0.68	0.75

43% of the Chinese predicate verbs are either not translated at all into English or are translated into a different part-of-speech category such as nouns or adjectives. As shown in Figure 4, the predicate verb 位/*located* in the input Chinese sentence is not translated in the system output

4 Semantic roles in SMT output

In the previous experiment, the semantic role accuracy in output translations was negatively affected by errors in identifying the central verb in the first place—as we have seen in both introductory examples of Section 3 as well as the example of Figure 1. Without the verb, properly identifying the arguments becomes meaningless. It is therefore worth asking a secondary version of the question: *providing the verb is correctly translated*, then how well do typical current SMT systems perform on semantic roles?

4.1 Experimental setup

Since nearly half of the predicate verbs in the system input are not translated or wrongly translated in the system output in the previous experiment, we construct another data set (data set B) by randomly selecting 50 bi-sentences under an additional constraint that all predicate verbs are correctly translated. We carry out the same analysis on data set B and the result is shown in Table 2.

4.2 Experimental results

For data set B, the overall F-measure of the translation of the semantic arguments is about 75%, which is 24 points higher than that in data set A.

In this data set B, we also find that some of the semantic roles are missing in the system output.

A common type of translation error occurs when a group of words that together have a single semantic role in the source language (Chinese) are split into separate groups in the translation (English) often in the wrong word order. In the example of Figure 5, the phrase 其所有资产的偿债率 in the input is translated into two separate phrases in the output: *its debt rate* and *of the assets*, creating different semantic relationships compared to the original semantic role of the source phrase. Finally, even though all the words in the arguments of a certain predicate verb are correctly translated into English, their semantic roles are found to be confusing in the translation leading to ambiguity in the interpretation of the translated sentences. As shown in the example of Figure 6, although words in both ARG0 and ARG1 are correctly translated into English, we still cannot understand the final translated sentence because the semantic roles of these two phrases are confused. We cannot tell which semantic roles *Myanmar and Thailand's government* and *the two countries border trade agreements* are supposed to play. This confusion arises from the incorrect position of the predicate verb *signed*.

As we can see, the types of translation errors shown in the examples of Figures 3 to 5 lead to ambiguity in the final understanding of the translation even though the system output still reads fluently. This is caused by the fact that current n-gram based SMT systems are not designed to take semantic roles into consideration.

5 Semantic vs. syntactic roles

The third experiment aims to answer another key question: if we favor semantic role consistency across both the source input sentence and the output translation, would this outperform merely favoring syntactic role consistency across the bisentence? In other words, does incorporating semantic role analysis contribute anything beyond the current work on syntactic SMT models?

5.1 Experimental setup

To address this question, we perform a different analysis of the previously described set of 745 bilingual sentence pairs with manually annotated syntactic and semantic roles from the Parallel PropBank.

The syntactic roles are manually annotated according to Treebank guidelines. Whereas the Chinese sentences are annotated with both “subject” and “object” syntactic roles, their English counterparts are only annotated with “subject” roles with-

IN [ARG0 上述 开发区] 基本 [PRED 位] 于 [ARG1 福建 经济 最为 活跃 的 东南部 地区] 。

REF The above-mentioned development zones are basically [PRED located] in the southeastern area of Fujian whose economy is the most active.

MT The basic development zones in the southeastern region of Fujian ' s economy is the most active.

Figure 4: Example of semantic frames in Chinese input and English MT output.

IN 去年四月， C R 公司开始了其破产程序，[ARG0 其所有资产的偿债率] 仅 [PRED 为] [ARG1 百分之五] 。

REF In April of last year, the CR Company began bankruptcy procedures and [ARG0 the debt compensation rate of all its assets] [PRED was] only [ARG1 5 %].

MT In April of last year, the company began bankruptcy procedures, all of its debt rate [PRED was] only [ARG1 five percent] [ARG0 of the assets] of the CR.

Figure 5: Example of semantic frames in Chinese input and English MT output.

Table 3: Syntactic role mapping in Chinese (ZH) to English (EN) translations.

Syntactic role mapping	**Freq**	**Pct**
ZH subject ↔ EN subject	514	84.26%
ZH subject ↔ EN NP	44	7.21%
ZH subject ↔ EN PP	31	5.08%
ZH subject ↔ EN S	15	2.46%
ZH subject ↔ EN other	6	0.98%

out the “object” roles.

Furthermore, we manually align the predicate argument structures across the bi-sentences for our experiment.

The experiment is done as follows:

1. We first extract all predicate argument structure mappings from the manually annotated and structurally aligned corpus. We compute the statistics of direct semantic role mappings (ARG*i* to ARG*i*) based on the translation.

2. From the output of step 1, we further look at the syntactic roles associated with each bilingual argument mapping. We use the semantic role boundaries from the annotated corpus to find the syntactic roles.

3. The corresponding Chinese/ English syntactic roles are then constructed as syntactic role mappings.

5.2 Experimental results

Given all the direct semantic role mappings from Chinese to English, their corresponding subject syntactic role mappings are listed below in Table 3. We can see that only 84.26% of direct semantic role mappings result from direct syntactic role projections. More than 15% of the subjects are not translated into subjects, even though their semantic roles are preserved across language.

This result shows that semantic roles enforce cross-lingual translation patterns more correctly than syntax. Whereas syntactic roles vary for each language, semantic roles that convey the meaning of a sentence are translingual.

6 Improving SMT with semantic frames

In the fourth experiment, we aim to assess the potential quantitative impact of realistic semantic role guidance to SMT systems, in terms of BLEU and METEOR scores. This is done by simulating the effect of enforcing consistency between the semantic predicates and arguments across both the input source sentence and the translation output.

6.1 Experimental setup

For this experiment, we return to data set B, as described in Section 4.1. For each sentence, two types of semantic parse based corrections are permitted to the output translation.

First, the constituent phrases corresponding to either the predicates or the arguments for any la-

IN [ARG0 缅甸 和 泰国 政府] 今天 下午 在 此间 [PRED 签订] 了 [ARG1 两 国 边境 贸易 协定]。

REF This afternoon [ARG0 the Myanmaran and Thai governments] [PRED signed] [ARG1 an agreement on border trade between their two countries] here.

MT [ARG? Myanmar and Thailand ' s government] of [ARG? the two countries border trade agreements] [PRED signed] here this afternoon.

Figure 6: Example of semantic frames in Chinese input and English MT output.

IN 加工贸易在广东外经贸发展中占有举足轻重的地位，同时也是粤港澳台经贸合作的重要内容。

REF The processing trade occupies a crucial position in the development of foreign economy and trade in Guangdong and at the same time is important content in the economic and trade cooperation between Guangdong , Hong Kong , Macao and Taiwan.

MT In the processing trade in Guangdong ' s foreign trade and economic development in Guangdong , Hong Kong , Macao , Taiwan occupies a decisive position at the same time , it is an important content of the economic and trade cooperation.

RE-ORDERED In the processing trade occupies a decisive position in Guangdong ' s foreign trade and economic development at the same time , it is an important content of the economic and trade cooperation in Guangdong , Hong Kong, Macao , Taiwan.

Figure 7: Example of semantic frames in Chinese input and English MT output.

Table 4: SMT performance improvement with semantic predicate and role consistency constraints.

Metric	Baseline translation	Enforcing consistent semantic parses
BLEU	34.76	36.62
METEOR	63.5	65.9

beled semantic role are permitted to be re-ordered such that a semantic parse of the re-ordered translation consistently matches the role label on the corresponding phrase in the input source sentence.

Second, if the translation of a predicate in the input source sentence is missing in the output translation, then a translation of that predicate may be added to the output translation such that, again, a semantic parse of the translation consistently associates it with the corresponding arguments for that predicate.

6.2 Experimental results

The results, as shown in Table 4, show that favoring semantic frame and role consistency across the source input sentence and the output translation improves BLEU and METEOR scores. The accuracy improves on the order of two points, for both metrics.

The example of Figure 7 shows how two of the constituent phrases are re-ordered.

It is worth noting that both BLEU and METEOR are still n-gram based metrics, which are of limited accuracy at evaluating fine-grained semantic distinctions in the translations. We suspect that the enhancement in translation quality would be even more obvious under utility-based MT evaluation strategies; this is one main direction for future research.

7 Conclusion

We have presented a series of experimental studies that illuminate more precisely the likely contribution of semantic roles in improving statistical machine translation accuracy. The experiments reported studied several aspects key to success: (1) the frequencies of types of SMT errors where semantic parsing and role labeling could help, and (2) if and where semantic roles offer more accurate guidance to SMT than merely syntactic annotation, and (3) the potential quantitative impact of realistic semantic role guidance to SMT systems, in terms of BLEU and METEOR scores. All sets of results support the utility of shallow semantic parsing and

semantic role labeling for improving certain limited but important aspects of SMT accuracy.

Our studies have focused on Chinese and English. Chinese and English are of course semantically very different, arising from their completely unrelated origins in the Sino-Tibetan and European language families. The effect is seen in the fact that state-of-the-art machine translation accuracy remains low for Chinese-English, even though intensive research on other "difficult" language pairs such as Arabic-English began far more recently. We conjecture that similar or better results to those reported in this paper would hold for most other language pairs, especially where there are closer correspondences in the semantic frame inventory of the two languages.

Acknowledgments We would like to thank Zhaojun Wu, Yongsheng Yang, Chi-kiu Lo, Huaqing Luo, Yihai Shen, and Marine Carpuat for their assistance. 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 GRF621008, GRF612806, DAG03/04.EG09, RGC6256/00E, and RGC6083/99E. 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.

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.

Marine Carpuat and Dekai Wu. 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 2005.

Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP– CoNLL 2007)*, pages 61– 72, Prague, Jun 2007.

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 and Dekai Wu, editors. *Proceedings of SSST–2, Second Workshop on Syntax and Structure in Statistical Translation, at ACL–08:HLT*. Association for Computational Linguistics, Columbus, OH, Jun 2008.

Pascale Fung, Zhaojun Wu, Yongsheng Yang, and Dekai Wu. Automatic learning of chinese–english semantic structure mapping. In *IEEE/ACL 2006 Workshop on Spoken Language Technology (SLT 2006)*, Aruba, Dec 2006.

Daniel Gildea and Dan Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):256–288, 2002.

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

Jesús Giménez and Lluís Màrquez. Linguistic features for automatic evaluation of heterogenous mt systems. In *Second Workshop on Statistical Machine Translation*, pages 256–264, Prague, Jun 2007.

Jesús Giménez and Lluís Màrquez. A smorgasbord of features for automatic mt evaluation. In *3rd ACL Workshop on Statistical Machine Translation*, pages 195– 198, Columbus, OH, Jun 2008.

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. Moses: Open source toolkit for statistical machine translation. In *ACL–2007 Demo and Poster Sessions*, pages 177–180, Prague, Jun 2007.

Franz Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. A smorgasbord of features for statistical machine translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL–2004)*, Boston, May 2004.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, Mar 2005.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei–Jing Zhu. BLEU: A method for automatic evaluation of machine translations. In *40th Annual Meeting of the Association for Computational Linguistics (ACL–2002)*, pages 311–318, Philadelphia, Jul 2002.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. Support vector learning for semantic argument classification. *Machine Learning*, 60(1–3):11–39, 2005.

Yihai Shen, Chi-kiu Lo, Marine Carpuat, and Dekai Wu. HKUST statistical machine translation experiments for IWSLT 2007. In *Fourth International Workshop on Spoken Language Translation (IWSLT 2007)*, pages 84–88, Trento, Oct 2007.

Honglin Sun and Daniel Jurafsky. Shallow semantic parsing of chinese. In *Human Language Technology Conference/ North American Chapter of the Association for Computational Linguistics (HLT/ NAACL– 2004)*, pages 249– 256, Boston, May 2004.

Dekai Wu and David Chiang, editors. *Proceedings of SSST–3, Third Workshop on Syntax and Structure in Statistical Translation, at NAACL–HLT 2009*. Association for Computational Linguistics, Boulder, CO, Jun 2009.

Dekai Wu. Stochastic Inversion Transduction Grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, Sep 1997.

Nianwen Xue and Martha Palmer. Automatic semantic role labeling for chinese verbs. In *19th International Joint Conference on Artificial Intelligence (IJCAI– 05)*, Edinburgh, 2005.

Nianwen Xue. Semantic role labeling of nominalized predicates in chinese. In *Human Language Technology Conference of the North American Chapter of the ACL (HLT–NAACL 2006)*, pages 431–438, New York, Jun 2006.

Are Unaligned Words Important for Machine Translation ?

Yuqi Zhang **Evgeny Matusov** **Hermann Ney**

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6 – Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{yzhang,matusov,ney}@cs.rwth-aachen.de

Abstract

In this paper, we deal with the problem of a large number of unaligned words in automatically learned word alignments for machine translation (MT). These unaligned words are the reason for ambiguous phrase pairs extracted by a statistical phrase-based MT system. In translation, this phrase ambiguity causes deletion and insertion errors. We present hard and optional deletion approaches to remove the unaligned words in the source language sentences. Improvements in translation quality are achieved both on large and small vocabulary tasks with the presented methods.

1 Introduction

Word alignment is a key part in the training of a statistical MT system because it provides mappings of words between each source sentence and its target language translation. Because of the difference in the structure of the involved languages, not all words in the source language have a corresponding word in the target language. So in the alignments, no matter manually created or automatically learned, some words are aligned, some are not.

Current state-of-the-art statistical machine translation is based on phrases. First the word alignments for the training corpus are generated. Then phrase alignments are inferred heuristically from the word alignments. This approach was presented by (Och et al., 1999) and implemented by e. g. (Koehn et al., 2003). Since this widely used phrase extraction method depends on word alignments, it is often assumed that the quality of word alignment is critical to the success of translation. However, some research have shown that the large gains in alignment accuracy often lead to, at best, minor gains in translation performance (Lopez and Resnik, 2006). They concluded that it could be more useful to directly investigate ways to reduce the noise in phrase extraction than improving word alignment. The work by (Ma et al., 2007) shows that a good phrase segmentation is important for translation result. Encouraged by the work, this paper explores the influence of the unaligned words on the phrase extraction and machine translation results. We show that the presence of unaligned words causes extraction of "noisy" phrases which can lead to insertion and deletion errors in the translation output. Furthermore, we propose approaches for "hard" and "soft" deletion of the unaligned words on the source language side. We then show that better way to deal with unaligned words can substantially improve translation quality, on both small and large vocabulary tasks.

In section 2, we briefly review the word alignment concept and point out that there is a large number of unaligned words in both manual and automatic alignments used for common translation tasks. In section 3, we explain how the unaligned words affect the phrase extraction and cause deletion and insertion errors. In section 4, we present two approaches to prove the negative impact of the unaligned words on translation quality. The experimental results are given in sections 5 and 6. Finally, section 7 presents a conclusion and future work.

2 Unaligned words in word alignment

In statistical translation models (Brown et al., 1990), a "hidden" alignment

Proceedings of the 13th Annual Conference of the EAMT, pages 226–233,
Barcelona, May 2009

$a_1^J := a_1, \ldots, a_j, \ldots, a_J$ is introduced for aligning the source sentence f_1^J to the target sentence e_1^I. The source word at position j is aligned to the target word at position $i = a_j$. The alignment a_1^J may contain special alignment $a_j = 0$, which means that the source word at index j is not aligned to any target word. Because a word in the source sentence cannot be aligned to multiple words in the target sentence, the alignment is trained in both translation directions: source to target and target to source. For each direction, a Viterbi alignment (Brown et al., 1993) is computed: $A_1 = \{(a_j, j)|a_j \geq 0\}$ and $A_2 = \{(i, b_i)|b_i \geq 0\}$. Here, a_1^J is the alignment from the source language to the target language and b_1^I is the alignment from the target language to the source language. To obtain more symmetrized alignments, A_1 and A_2 can be combined into one alignment matrix A with the following combination methods. More details are described in (Och and Ney, 2004):

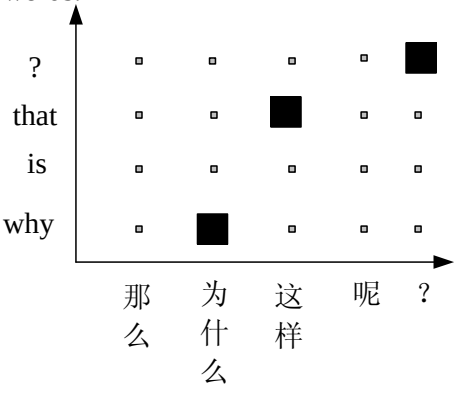
- $intersect$: $A = A_1 \cap A_2$
- $union$: $A = A_1 \cup A_2$
- $refined$: extend from the intersection. $intersect \subseteq refined \subseteq union$

In any of the alignments, there are many words which are unaligned. We have counted unaligned words in various Chinese-English alignments both a small corpus (LDC2006E93[1]) and a large corpus (GALE-All[2]). Table 1 presents what percentage of unaligned words occurs in each alignment. Since the released LDC2006E93 corpus contains manual alignments, we can see that even in "correct" alignments, more than 10% words are unaligned. $intersect$, the alignment with the best precision, has around 50% unaligned words on both sides. IN $union$, which has best recall, still around 10% of the words are unaligned. The most often used $refined$ alignment, which has the balance between precision and recall, has about 25% unaligned words. Since phrase pairs are extracted from the word alignments, these unaligned words will affect them as described below.

[1]LDC2006E93: LDC GALE Y1 Q4 Release - Word Alignment V1.0, Linguistic Data Consortium (LDC)

[2]GALE-ALL: all available training data for Chinese-English translation released by LDC. http://projects.ldc.upenn.edu/gale/data/DataMatrix.html

Figure 1: An alignment example with unaligned words.



为什么	why
为什么	why is
那么 为什么	why
那么 为什么	why is
为什么 这样	why is that
为什么 这样 呢	why is that
那么 为什么 这样	why is that
那么 为什么 这样 呢	why is that
为什么 这样 呢 ？	why is that ?
那么 为什么 这样 呢 ？	why is that ?
那么	is
呢	is
这样	is that
这样	that
这样 呢	is that
这样 呢	that
这样 呢？	is that ?
这样 呢？	that ?
呢 ？	?
？	?

3 Phrase extraction

In the state-of-the-art statistical phrase-based models, the unit of translation is any contiguous sequence of words, which is called a phrase. The phrase extraction task is to find all bilingual phrases in the training data which are consistent with the word alignment. This means that all words within the source phrase must be aligned only with the words of the target phrase; likewise, the words of the target phrase must be aligned only with the words of the source phrase (Och et al., 1999) (Zens et al., 2002). A target phrase can have multiple consistent source phrases if there are unaligned words at the boundary of the source phrase and vice versa.

Figure 1 gives an alignment example with un-

Corpus	Sentence	Alignment	Unaligned Chinese words	Unaligned English words
LDC2006E93	10,565	$manual$	14%	11%
		$intersect$	53%	40%
		$refined$	23%	23%
		$union$	7%	14%
GALE-All	8,778,755	$intersect$	48%	55%
		$refined$	24%	27%
		$union$	9%	16%

Table 1: The percentages of unaligned words in variant alignments.

aligned words on both source and target sides and the phrase table extracted from this alignment. The unaligned words will result in multiple extracted phrase pairs. All of these phrase pairs are kept because the unaligned words are necessary to complete a good sentence though they have no corresponding translations. However, the translation models are not powerful enough to select the correct phrase pair from these multiple pairs. As a result, this ambiguity often causes insertion errors which is adding redundant words to the translation and deletion errors which means that translations of some source words are missing. We have used the phrase table in figure 1 to translate the source sentence. (The translation system will be described in the section 5). Since the example sentence is short, to see how the phrase pairs are concatenated, we limit the length of the used phrase from 1 to 4. In the table 2 there is an insertion error with $slen = 1$, $tlen = 1$, which is caused by the unaligned 'is' in the phrase '呢# is'. With $slen = 2$, $tlen = 2$ and $slen = 3$, $tlen = 3$ there are deletion errors where unaligned 'is' is missing in phrase '那么 为什么why is# is'.

4 Deletion of the unaligned words in source sentences

Based on the observations in the last section, we are going to disambiguate the multiple phrase pairs caused by unaligned words. In the automatically trained alignment there are a few possible cases for the unaligned words.

correct vs. wrong: an unaligned word is correct if it really has no corresponding translations and is left unaligned by a human annotator. An unaligned word is wrong if it has been aligned in the manual alignment.

function words vs. content words: Comparing the alignment of function words and content words, we could find that the correct unaligned words are roughly function words, while the wrong unaligned words are usually content words. The function words have little lexical meaning, but instead serve to express grammatical relationships with other words within a sentence On the contrary, the content words usually carry meaning, which are "natural units" of translation between languages.

If we just focus on the disambiguation of multiple phrases and not consider applying grammatical information in function words to the translation system, like the work done by (Setiawan et al., 2007), the simplest way of reducing the multiple phrases is to delete the 'correct' unaligned words: the function words. The function words at the target side should not be touched, since they are necessary to complete a good sentence. However, the function words at the source side could be removed, when they have no corresponding translations.

4.1 Deletion Candidates

Not all unaligned words should be removed. Besides the content words, a source function word could also have correct mappings to the target words in some sentences. We have used two constraints to filter out the words which can be deleted..

We use relative frequencies to estimate the probability of a word being aligned.

$$p(w_{align}) = \frac{N_{w_{align}}}{N(w)} \quad (1)$$

The number of times a word w is aligned in the training data is denoted by $N_{w_{align}}$, and $N(w)$ is the total number of occurrences of the word w. The

slen=1 tlen=1	why # 为什么 ## is # 那么 ## that # 这样 ## is # 呢 ## ? # ？	why is that is ?
slen=2 tlen=2	why # 那么 为什么 ## that # 这样 ## ? # 呢 ？	why that ?
slen=3 tlen=3	why # 那么 为什么 ## that ? # 这样 呢 ？	why that ?
slen=4 tlen=4	why is that # 那么 为什么 这样 ## ? # 呢 ？	why is that ?

Table 2: The translations of the example with phrase length limitation. The symbol ## denotes concatenation of phrase pairs.

first constraint is that the probability of a word being aligned is below a threshold τ.

$$Con\_p(w) = \begin{cases} 1 & \text{if } p(w_{align}) \leq \tau \\ 0 & \text{if } p(w_{align}) > \tau \end{cases} \quad (2)$$

This constraint can be used with different thresholds. The smaller the threshold is, the more strict constraint is applied and fewer words are to be considered. When $p(w_{align})$ is 0.5, it means that the word has the same probability to be aligned and not to be aligned. In order to filter out the deletion candidates, the best threshold as determined in our experiments should be less than 0.5.

The second constraint is to use the POS tags to mark the function words. In general, the content words include nouns, verbs, adjectives, and most adverbs. We denote the POS tag set for content words as $S = \{noun, verb, adj, adv\}$. The constraint for the function word is:

$$Con\_fun(w) = \begin{cases} 1 & \text{if } POS(w) \notin S \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In the experiments, we will test both $Con\_p(w)$ and $Con\_p(w)$+$Con\_fun(w)$. We will show that it is more important for a deletion candidate to be constrained by $Con\_p(w)$, since content and function words in linguistics are not always distinguished clearly.

4.2 Hard deletion

The simplest way of deletion is directly removing the found words from the source sentences and the alignments. The change of the alignment will affect not only the extracted phrase pairs around the deleted word, but also the probability estimation of all phrases. In this way, the source sentences become relatively shorter. The size of the phrase table will be smaller because of the reduction in the multiple translation pairs. However, the drawback of the method is obvious. Most words are aligned or not in different contexts. When we set τ greater than 0 and delete the filtered words, there must be some words which should actually be translated, which means that they were deleted wrongly.

Hard deletion is an easy method to investigate the influence of unaligned words on translation results. Although the method will cause overdeletion, it can reflect which multiple translation pairs containing an unaligned word provide more useful information or more harmful information for ultimate translation quality.

4.3 Optional deletion

A better and more complicated method is to apply optional deletion. We do not make a firm decision to delete any words. Instead, we preserve ambiguity and defer the decision until later stages.

We use a confusion network (CN) to represent the ambiguous input. Some works are reported to use CNs in machine translation (Bertoldi et al., 2007; Koehn et al., 2007). A CN is a directed acyclic graph in which each path goes through all the nodes from the start node to the end node. Its edges are labeled with words. An example of a CN for optional deletion is shown in table 3.

把.38 ε.62	机票1.0	忘1.0	在1.0	家里1.0	了.21 ε.79

Table 3: A CN example of optional deletion.

The special empty-word ε represents a word deletion. Also, the word aligned probability is attached to each edge. The probability is calculated by equation (1). When the word is a content word, its aligned probability is 1.0. The score with $epsilon$ means the probability of the word in the same column not to be aligned, which is equal to $1 - p(w_{align})$.

Input source sentences are represented by CN. Like what is done in the hard deletion, the alignments are modified by deleting all deletion candidates and the corresponding points in the alignment matrix. However, to match the possible non-deletion of the unaligned words, the original alignment is also needed. We combine the two alignments by merging the phrase counts and recompute the phrase probabilities.

5 Experimental Setup

5.1 Data

We carried out MT experiments for translation from Chinese to English on two data sets: BTEC08 and GALE08.

The BTEC08 data was provided within the IWSLT 2008 evaluation campaign (Paul, 2008), extracted from the Basic Traveling Expression Corpus(BTEC) (Takezawa et al., 2002). The data is a multilingual speech corpus which contains sentences which are usually found in books for tourists. The sentences are short, with less than 10 words on average. The parallel training data is relatively small. We added the official IWSLT08 training data, the IWSLT06 dev data and IWSLT06 evaluation data and their references to the training data. The development and test sets in the experiments below are from the IWSLT04 and IWSLT05 evaluation data. We found that the two data sets are not similar, so we took the first half of each and combine them as dev data. The remaining two halves are combined as test data.

The large vocabulary GALE data were all provided by LDC. The test data has four genres: broadcast news (BN), broadcast conversations (BC), newswire (NW) and webtext (WT). The first two genres are for speech translation and the last two are for text translation. Here, we only carried out experiments on NW. The sentences of the GALE task are longer (around 30 words per sentence) and more difficult to translate.

The corpus statistics for both tasks are shown in Table 4:

BTEC		Chinese	English
Train:	Sentences	23940	
	Running words	181486	232746
Dev:	Sentences	503	
	Running words	3085	3887
Test:	Sentences	503	
	Running words	3109	3991
GALE		Chinese	English
Train:	Sentences	8778755	
	Running words	232799466	249514713
Dev08:	Sentences	485	
	Running words	14750	16570
Test08:	Sentences	480	
	Running words	14800	16683

Table 4: Corpus Statistics of the BTEC and GALE translation tasks. For BTEC dev and test sets, the number of English tokens is the average over 16 human reference translations.

5.2 Baseline System

Our baseline system is a standard phrase-based SMT system. Word alignments are obtained by using GIZA++ (Och and Ney, 2003) with IBM model 4 [3]. We symmetrized bidirectional alignments using the refined heuristic (Och and Ney, 2004). The phrase-based translation model is a log-linear model that include phrase translation probabilities and word-based translation probabilities in both translation directions, phrase count models, word and phrase penalty, target language model (LM) and a distortion model. Language models were built using the SRI language modeling toolkit (Stolcke, 2002). On the small vocabulary BTEC task we used a 6-gram. On the large vocabulary GALE task we included 5-gram language model probabilities. The model scaling factors are optimized on the development set with the goal of improving the BLEU score. We used a non-monotonic phrase-based search algorithm that can take confusion networks as input in the style of (Matusov et al., 2008).

[3]Specifically, on GALE data we performed 5 iterations of Model 1, 5 iterations of HMM, 2 iterations of Model 4. On BTEC data we performed 4 iterations of Model 1, 5 iterations of HMM, 8 iterations of Model 4.

6 Experimental Results

6.1 The deletion candidates

First, we tested different thresholds τ in the range from 0.2 to 0.5. The set with small τ is a subset of the one with larget τ. We filtered out words which were most frequently not aligned in training. We performed the experiments on both BTEC and GALE tasks. The findings are reported in table 5. For each threshold the table gives the number of unique words removed (num.) and some examples.

By applying the two constraints Con_p+Con_fun, the number of deletion candidates is reduced greatly. That means among the unaligned words in alignments there are many content words. The content words, especially nouns, usually are expected to be translated. It is not good if there are many content words unaligned.

Comparing between BTEC and GALE there are fewer deletion candidates in GALE data, both con-

	BTEC				GALE			
	Con_p		Con_p+Con_fun		Con_p		Con_p+Con_fun	
τ	num.	example	num.	example	num.	example	num.	example
0.2	1	的	1	的	1	恭	0	-
0.3	4	的 了 哭 却	3	的 了 却	7	的 慧 毛病 ...	1	的
0.4	21	叶 以 把 战争 ...	10	以 把 ...	17	的 罗斯 ...	1	的
0.5	152	呀 对 着 当时 ...	20	呀 对 着 ...	62	的 中 之 兆...	3	的 中 之

Table 5: Some statistics and examples of the words removed based on the constraints defined in equations 2 and 3.

tent and function words. It implies that large data leads to obtain better alignments which assign more mappings between source and target languages.

6.2 Hard deletion

Since the hard deletion is easy to carry out, we performed the experiments on both BTEC and GALE tasks here, too. As the number of deletion candidates on GALE is small, we tested the smallest deletion candidate set "的" and the biggest set which is under the constraint Con_p with $\tau = 0.5$. Translation results are shown in table 6. The second row "rm-1" is the hard deletion of 的 and the third row "rm-62" is for the deletion of the 62 words as shown in table 5.

It is interesting to see that the deletions of both the small set and large set of words improve the baseline on every metric. 的 is the most common function word in Chinese to connect adjectives and nouns and it is also the word with lowest aligned probability in the table 5. The BLEU and TER scores both improve 0.5% absolute on dev and test data just by removing this single word. However, when we remove the 62 words including 的, the result does not improve further. This means that the deletion candidate set contains some content words, the deletion of which has a negative influence on translation quality.

The BTEC data provides us with a larger deletion candidate set. Additionally, the small size of the training data for the BTEC task makes it possible to run some finer-grained experiments. We focus on how the removable function words affect the translation quality. The experiments are carried on the word set with different thresholds τ and under the constraint Con_p+Con_fun. The translation results with hard deletion on BTEC are shown in table 7.

The improvement in the BTEC data is not as much as on the GALE data. Only when τ is set to 0.4, we obtained slightly better scores. The reason is that extracted phrases are very long comparing to the sentence length. The maximum phrase length was set to 15 words, both for BTEC and GALE task. However, the average sentence length of the BTEC test set is around 7 words, vs. 30 words on the GALE task. When phrase pairs are longer, there are fewer cases that unaligned words are at their boundaries. The translation examples in table 2 also reflect this phenomenon. That source sentence has 5 words. When the phrase length limitation is 4, unaligned 'is' is an inner word in the phrase pair *why is that #* 那么 为什么 这样.

6.3 Optional deletion

In addition to the hard deletion experiments on BTEC, we carried out the optional deletion experiments in the same settings. The results are also shown in table 7. The optional deletion method achieved good performance. The BLEU score improves consistently with all settings, at most 1.5% on the dev set and 0.7% on the test set with $\tau = 0.4$.

Furthermore, we are also interested in the influence of individual deletion candidate on the translation results. It would be more useful if we know what words are important for the deletion instead of just determining the optimal threshold. Since $\tau = 0.4$ has achieved the best result both in hard deletion and optional deletion, we explore the 10 removable function words in the set one by one. The 10 words are listed in table 8. At first, we sorted the 10 words according to the probability of being aligned. From the low to high probability, we add one word a time to the deletion candidate set. The results are shown in table 8. The word 的, which has the lowest probability of being aligned, is the most important word in the set.

	dev08				test08			
%	BLEU	Interval	TER	Interval	BLEU	Interval	TER	Interval
baseline	31.5	[30.4, 32.7]	60.7	[59.9, 61.5]	30.9	[29.8, 32.0]	60.3	[59.5, 61.1]
rm-1:的	31.9	[30.6, 33.1]	60.2	[59.1, 61.2]	31.4	[30.3, 32.7]	59.7	[58.9, 60.7]
rm-62	32.3	[31.0, 33.6]	60.1	[59.0, 61.0]	31.2	[30.0, 32.3]	59.9	[59.0, 60.8]

Table 6: Translation results using the hard deletion method on the GALE task.

	dev				test			
%	BLEU	Interval	TER	Interval	BLEU	Interval	TER	Interval
baseline	49.6	[47.0, 52.6]	41.3	[39.1, 43.5]	49.5	[46.8, 52.1]	41.3	[39.3, 43.3]
rm-funW	Hard deletion							
$\tau = 0.2$	49.1	[46.3, 52.1]	41.9	[39.6, 43.9]	49.7	[47.0, 52.3]	41.5	[39.5, 43.6]
$\tau = 0.3$	50.0	[47.1, 52.9]	41.0	[38.8, 43.5]	49.3	[46.4, 51.9]	41.2	[39.3, 43.6]
$\tau = 0.4$	50.0	[46.9, 52.9]	41.3	[39.4, 43.8]	49.7	[47.1, 52.6]	41.1	[39.0, 43.3]
rm-funW	Optional deletion							
$\tau = 0.2$	51.1	[48.6, 54.2]	40.5	[38.2, 42.7]	49.6	[46.7, 52.6]	41.5	[39.3, 43.7]
$\tau = 0.3$	51.2	[48.9, 53.9]	40.4	[38.5, 42.1]	49.9	[47.1, 52.6]	41.5	[39.5, 43.8]
$\tau = 0.4$	51.1	[48.5, 53.5]	40.6	[38.7, 42.9]	50.2	[47.7, 53.0]	41.4	[39.3, 43.5]

Table 7: Translation results using hard and optional deletion methods on the BTEC task.

We also calculate the 95% confidence intervals for both hard deletion and optional deletion. Unfortunately, the new systems are not statistical significant though the BLEU scores are better.

7 Conclusion and future work

In this paper, we have devoted attention to the problem of a large number of unaligned words in the word alignments generally used for MT model training. These unaligned words result in ambiguous phrase pairs being extracted by a state-of-the-art phrase-based statistical MT system. In translation, this phrase ambiguity causes deletion and insertion errors. We classified the unaligned words into function words and content words and showed that unaligned function words have an important influence on phrase extraction.

Furthermore, we have proposed two methods to improve phrase extraction based on handling of unaligned words. Since it is important to keep the unaligned words on the target side to obtain complete and fluent translations, we have applied hard deletion and optional deletion of the unaligned words on the source side before phrase extraction. Though the methods are simple, they still achieved notable improvements in automatic MT evaluation measures on both small and large vocabulary tasks. We have shown that differentiating between useful and "removable" unaligned words is important for the quality of the extracted phrases and, consequently, for the quality of the phrase-based MT.

This paper pointed out the importance of unaligned words, but only considered the source language words. In the future, more work should be done regarding the unaligned words in the target language. The translations are more directly affected by the quality of target phrases. Since deleting of unaligned words at the target side is clearly not the right solution, some disambiguation models are to be investigated.

8 Acknowledgments

This material is partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023 and partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

References

P.F.Brown, J.Cocke, S.A.Della Pietra, V.J.Della Pietra, F. Jelinek, J.D.Lafferty, R.L.Mercer, and P.S. Roossin. 1990. A statistical approach to machine translation., In *Computational Linguistics,16(2)*, pages 79-85, Jun.

P.F.Brown, S.A.Della Pietra, V.J.Della Pietra, and R.L.Mercer. 1993. The mathematics of statistical

		Hard deletion		optional deletion	
		dev	test	dev	test
	$p(w_{align})$	BLEU[%]	BLEU[%]	BLEU[%]	BLEU[%]
baseline	-	49.6	49.5	49.6	49.5
的	0.007	49.1	49.7	51.1	49.6
+ 了	0.21	50.0	49.3	51.2	49.9
+ 却	0.27	50.0	49.3	51.2	49.9
+ 以	0.35	50.0	49.4	51.2	49.9
+ 把	0.38	50.0	49.7	51.1	50.2
+ 对于	0.4	50.1	49.7	51.1	50.1
+ 既	0.4	50.1	49.6	51.1	50.1
+ 着	0.4	50.1	49.7	51.1	50.1
+ 式	0.4	50.1	49.6	51.1	50.1
+ 对	0.4	50.0	49.7	51.1	50.2

Table 8: The influence of deleting individual words on the translation quality (BTEC task).

machine translation: Parameter estimation., In *Computational Linguistics,19(2)*, pages 263-311

Nicola Bertoldi, Richard Zens and Marcello Federico. 2007. Speech Translation by Confusion Network Decoding, In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1297-1300, Apr.

Philipp Koeln, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation, In *Proceedings of the 2003 Human Language Technology onference of the North American Chapter of the Association for Computational Linguistics(HLT-NAACL)*, pages 127-133, May.

Philipp Koehn, Nicola Bertoldi, Ondrej Bojar, Chris Callison-Burch, Alexandra Constantin, Brooke Cowan, Chris Dyer, Marcello Federico, Evan Herbst, Hieu Hoang, Christine Moran, Wade Shen, and Richard Zens. 2007. Open Source Toolkit for Statistical Machine Translation: Factored Translation Models and Confusion Network Decoding, *CLSP Summer Workshop Final Report WS-2006*,

Adam Lopez and Philip Reesnik. 2006. Word-Based Alignment, Phrase-Based Translation: What's the Link?, In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 90-99, Aug.

Evgeny Matusov and Björn Hoffmeister and Hermann Ney. 2008. ASR Word Lattice Translation with Exhaustive Reordering is Possible, In *Proceedings of the Interspeech 2008*, pages 2342-2345, Sep.

Yanjun Ma and Nicolas Stroppa and Andy Way 2007. Bootstrapping Word Alignment via Word Packing In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 304–311, Jun. Prague, Czech Republic.

Franz Josef Och, Christoph Tillman, and Hermann Ney. 1999. Improved alignment models for statistical machine translation, In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methos in Natural Language Processing and very Large Corpora(EMNLP-VLC)*, pages 20-28, Jun.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models, In *Computational Linguistics,29(1):19-51*

Franz Josef Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation, In *Computational Linguistics*

Michael Paul. 2008. Overview of the IWSLT 2008 evaluation campaign, In *Proceedings of the International Workshop on Spoken Language Translation*, pages 1–17, Oct.

Andrea Stolcke. 2002. SRILM - An extensible language modeling toolkit, In *Proceedings of the International Conference on Spoken Language Processing*, pages 901-904.

Hendra Setiawan and Min-Yen Kan and Haizhou Li. 2007. Ordering Phrases with Function Words, In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* pages 712–719, Jun.

Toshiyuki Takezawa and Eiichiro Sumita and Fumiaki Sugaya and Hirofumi Yamamoto and Seiichi Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world, In *Proceedings of Third International Conference on Language Resources and Evaluation 2002 (LREC)*, pages 147-152

Richard Zens, Franz Josef Och and Hermann Ney. 2002. Phrase-based statistical machine translation, In *25th German Conference on Artificial Inteligence (LNAI)*, pages 18-32, Sep.

Using Supertags as Source Language Context in SMT

Rejwanul Haque[†], **Sudip Kumar Naskar**[†], **Yanjun Ma**[*] and **Andy Way**[†*]
CNGL[†]/NCLT[*]
School of Computing
Dublin City University
Dublin 9, Ireland
{rhaque, snaskar, yma, away}@computing.dcu.ie

Abstract

Recent research has shown that Phrase-Based Statistical Machine Translation (PB-SMT) systems can benefit from two enhancements: (i) using words and POS tags as context-informed features on the source side; and (ii) incorporating lexical syntactic descriptions in the form of supertags on the target side. In this work we present a novel PB-SMT model that combines these two aspects by using supertags as source language context-informed features. These features enable us to exploit source similarity in addition to target similarity, as modelled by the language model. In our experiments two kinds of supertags are employed: those from Lexicalized Tree-Adjoining Grammar and Combinatory Categorial Grammar. We use a memory-based classification framework that enables the estimation of these features while avoiding problems of sparseness. Despite the differences between these two approaches, the supertaggers give similar improvements. We evaluate the performance of our approach on an English-to-Chinese translation task using a state-of-the-art phrase-based SMT system, and report an improvement of 7.88% BLEU score in translation quality when adding supertags as context-informed features.

1 Introduction

In log-linear phrase-based SMT, the probability $P(e^I_1|f^J_1)$ of a target phrase e^I_1 given a source phrase f^J_1 is modelled as a log-linear combination of features which normally consist of a finite set of translational features, and a language model (Och and Ney, 2002). The usual translational features involved in those models express dependencies between the source and target phrases, but not dependencies between the phrases in the source language themselves. Stroppa et al. (2007) were the first to show that incorporating source language context using neighbouring words and part-of-speech tags had the potential to improve translation quality.

In a separate strand of research, Hassan et al. (2006, 2007, 2008) showed that incorporating lexical syntactic descriptions in the form of supertags in the target language model and on the target side of the translation model could improve significantly on state-of-the-art approaches to MT. Despite the significance of this work, it is currently not possible to develop a fully supertagged PB-SMT system given that supertaggers exist only for English.

In this paper, we begin to explore whether such a system could indeed generate improvements across all PB-SMT system components. Our novel approach combines the methods of (Stroppa et al., 2007) and (Hassan et al., 2006, 2007, 2008; Hassan, 2009) in one model. We extend a standard PB-SMT system with syntactic descriptions on the source side. Crucially, the kind of lexical descriptions that we employ are those that are commonly devised within lexicon-driven approaches to linguistic syntax, namely Lexicalized Tree-Adjoining Grammar (LTAG: Joshi and Schabes, 1992; Bangalore and Joshi, 1999) and Combinatory Categorial Grammar (CCG: Steedman, 2000). In such approaches, the grammar consists of a very rich lexicon and a small set of combinatory operators that assemble lexical entries together into parse-trees. The lexical entries consist of syntactic constructs ('supertags') that describe information such as the POS tag of the word, its subcategorisation information and the hierarchy of phrase categories

Proceedings of the 13th Annual Conference of the EAMT, pages 234–241,
Barcelona, May 2009

that the word projects upwards. Like (Hassan et al., 2006, 2007, 2008; Hassan, 2009), in this work we employ the lexical entries but exchange the algebraic combinatory operators with the more robust and efficient supertagging approach: like standard taggers, supertaggers employ probabilities based on local context and can be implemented using finite state technology, e.g. Hidden Markov Models (Bangalore and Joshi, 1999).

There are currently two supertagging approaches available: LTAG-based (Bangalore and Joshi, 1999) and CCG-based (Clark and Curran, 2004). Both the LTAG (Chen et al., 2006) and the CCG supertag sets (Hockenmaier, 2003) were acquired from the WSJ section of the Penn-II Treebank using hand-built extraction rules. Here we test both the LTAG and CCG supertaggers. We extract the supertagged components of context words (±1/±2) along with the source phrase (Koehn et al., 2003) in a standard PB-SMT system. We use a memory-based classification approach to obtain the probability for the given additional contexts with the source phrase. In this paper we discuss these and other empirical issues.

The remainder of the paper is organized as follows. In section 2 we discuss related work. Section 3 gives a brief overview of PBSMT. In section 4 we describe the context-informed features contained in our baseline log-linear phrase-based SMT system. In section 5 we describe the memory-based classification approach. Section 6 describes the features used in the experiments, and the pre-processing required. Section 7 includes the results obtained, together with some analysis. Section 8 concludes, and provides avenues for further work.

2 Related Work

(Berger et al., 1996) first suggested context-sensitive modelling of word translations in order to integrate local contextual information into their IBM translation models using a Maximum Entropy (MaxEnt) model, but the work is not supported by any significant evaluation results.

García Varea et al. (2001) present a MaxEnt approach to integrate contextual dependencies into the EM algorithm of the statistical alignment model to develop a refined context-dependent lexicon model. Using such a model on the German—English Verbmobil corpus, they obtained better alignment quality in terms of improved alignment error rate (AER). However, since alignment is not an end task in itself and most often used as an intermediate task to generate phrase pairs for the t-tables in PB-SMT systems, improved AER scores do not necessarily result in improved translation quality, as noted by a number of researchers.

(Vickrey et al., 2005) built classifiers inspired by those used in word-sense disambiguation (WSD) to fill in any blanks in a partially completed translation. (Giménez and Màrquez, 2007) extended this work by considering the slightly more general case of very frequent phrases and moved to full translation rather than blank-filling on the target side.

Initial attempts to embed context-rich approaches from WSD methods into SMT systems to enhance lexical selection did not lead to any improvement in translation quality (Carpuat and Wu, 2005). However, more recent approaches (Carpuat and Wu, 2007; Chan et al., 2007; Giménez and Màrquez, 2007) of integrating state-of-the-art WSD methods into SMT to improve the overall translation quality have met with more success.

Language models arguably play the most significant role in today's PB-SMT systems. It is obvious that a straightforward addition of a source language model will make no contribution as this will be cancelled out by the denominator in the noisy-channel model of SMT. However, for some time now the feeling was that some incorporation of source language information into SMT systems had to help. (Stroppa et al., 2007) added source-side contextual features to a state-of-the-art log-linear PB-SMT system by incorporating context-dependent phrasal translation probabilities learned using decision trees. They considered up to two words and/or POS tags on either side of the source focus word as contextual features. In order to overcome problems of estimation of such features, they used a decision-tree classifier which implicitly smoothes the probability estimates. Significant improvements over a baseline state-of-the-art PB-SMT system were obtained on Italian—English and Chinese—English IWSLT tasks.

Unlike other recent proposals to exploit the accuracy and the flexibility of discriminative learning (e.g. Cowan et al., 2006; Liang et al., 2006), the strength of the approach of (Stroppa et al., 2007) is that no redefinition of one's training procedures is required.

Like the work of (Max et al., 2008), the present work is directly motivated by and an extension of the approach of (Stroppa et al., 2007).

The work of both (Max et al., 2008) and (Gimpel and Smith, 2008) focus on language pairs where the target is not English. While (Gimpel and Smith, 2008) are unable to show any improvements for English→German, (Max et al., 2008) conduct experiments from English→French. Using the same sorts of local contextual features as (Stroppa et al., 2007), as well as using broader context in addition to grammatical dependency information, (Max et al., 2008) show modest gains over a PB-SMT baseline model in terms of automatic evaluation scores, but more improvements come to light in a manual investigation.

One final paper in this strand of research is that of (He et al., 2008), who despite not mentioning the obvious link between the two pieces of work, show that the source language features used by (Stroppa et al., 2007) are also of benefit when used with the Hiero (Chiang, 2007) decoder.

As regards supertagged models of translation, (Hassan et al., 2006, 2007b, 2008; Hassan, 2009) have demonstrated clearly that adding supertags (essentially, part-of-speech tags of words plus local subcategorisation requirements) in the target language model and on the target side of the translation model improve state-of-the-art PB-SMT systems. The system of (Hassan et al., 2007a) was ranked first according to human evaluators on the IWSLT 2007 Arabic–English task, despite the improvements in system design not being shown to their best advantage by the automatic evaluation metrics. More recently, (Hassan, 2009) has demonstrated that improvements can even be gained over the leading NIST-07 Arabic–English system of (Ittycheriah and Roukos, 2007).

3 Log-Linear PB-SMT

Translation is modelled in PB-SMT as a decision process, in which the translation $e_1^I = e_1 \ldots e_I$ of a source sentence $f_1^J = f_1 \ldots f_J$ is chosen to maximize (1):

$$\arg\max_{I,e_1^I} P(e_1^I \mid f_1^J) = \arg\max_{I,e_1^I} P(f_1^J \mid e_1^I).P(e_1^I) \quad (1)$$

where $P(f_1^J \mid e_1^I)$ and $P(e_1^I)$ denote respectively the translation model and the target language model (Brown et al., 1993). In log-linear phrase-based SMT, the posterior probability $P(e_1^I \mid f_1^J)$ is directly modelled as a (log-linear) combination of features (Och and Ney, 2002), that usually comprise M translational features, and the language model, as in (2):

$$\log P(e_1^I \mid f_1^J) = \sum_{m=1}^{m} \lambda_m h_m(f_1^J, e_1^I, s_1^K) + \lambda_{LM} \log P(e_1^I) \quad (2)$$

where $s_1^k = s_1 \ldots s_k$ denotes a segmentation of the source and target sentences respectively into the sequences of phrases $(\hat{e}_1,\ldots,\hat{e}_k)$ and $(\hat{f}_1,\ldots,\hat{f}_k)$ such that (we set $i_0 = 0$) (3):

$$\forall 1 \le k \le K,\ s_k = (i_k; b_k, j_k),$$
$$\hat{e}_k = e_{i_{k-1}+1} \ldots e_{i_k},$$
$$\hat{f}_k = f_{b_k} \ldots f_{j_k} \quad (3)$$

The translational features involved depend only on a pair of source/target phrases and do not take into account any context of these phrases. This means that each feature h_m in (2) can be rewritten as in (4):

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) \quad (4)$$

where $\hat{h}_m$ is a feature that applies to a single phrase-pair. It thus follows:

$$\sum_{m=1}^{m} \lambda_m \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) = \sum_{k=1}^{K} \hat{h}(\hat{f}_k, \hat{e}_k, s_k) \quad (5)$$

where, $\hat{h} = \sum_{m=1}^{m} \lambda_m \hat{h}_m$. In this context, the translation process amounts to: (i) choosing a segmentation of the source sentence, (ii) translating each source phrase, and (iii) re-ordering the target segments obtained.

4 Source Context Features in Log-Linear PB-SMT

As well as using local words and POS-tags as features, as in (Stroppa et al., 2007), we introduce supertags as a syntactic source context feature in the log-linear model of PB-SMT. The context of a source phrase $\hat{f}_k$ is defined as the sequence before and after a focus phrase $\hat{f}_k = f_{i_k} \ldots f_{j_k}$. In the following sections we describe both the lexical and syntactic features used.

4.1 Lexical Context Features

These features include the direct left and right context words of length l (resp. $f_{i_k-1} \ldots f_{i_k-l}$

and $f_{j_k+1}...f_{j_k+l}$) of a given focus phrase $\hat{f}_k = f_{i_k}...f_{j_k}$. It forms a window of size $2l+1$ features including the focus phrase. Thus lexical contextual information (CI) can be described as in (6):

$$\text{CI} = \{ f_{i_k-l}...f_{i_k-1}, \hat{f}_k, f_{j_k+1}...f_{j_k+l} \} \quad (6)$$

In our experiments we used ±1 and ±2 context words (i.e. l=1, 2).

4.2 Syntactic Context Features

We considered the syntactic information (SI) of the focus phrase and of the context words. The syntactic information we use are supertags and/or POS tags. In our model, the supertag or POS tag of a multi-word focus phrase is the concatenation of the supertags or POS tags of the words composing that phrase. We can thus describe our syntactic contextual information as in (7):

$$\text{CI} = \{SI(f_{i_k-l})...SI(f_{i_k-1}), \hat{f}_k, SI(\hat{f}_k), SI(f_{j_k+1})...SI(f_{j_k+l})\} \quad (7)$$

Thus a window of size $2l+2$ features is formed including the focus phrase and syntactic information of that phrase. In our experiments we used ±1 and ±2 syntactic information (i.e. l=1, 2). We also experimented with both supertag and POS tag features to see whether further improvements could be found. In such cases the contextual information is formed by the union of the two syntactic features, i.e. $\text{CI} = \text{CI}_{\text{syn1}} \cup \text{CI}_{\text{syn2}}$. We can also combine the syntax and the lexical contextual information in a similar way, if required.

One natural way of expressing a context-informed feature is as the conditional probability of the target phrase given the source phrase and its context information, as in (8):

$$h_m(f_k, CI(\hat{f}_k), \hat{e}_k, s_k) = \log P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k)) \quad (8)$$

5 Memory-Based Classification

As (Stroppa et al., 2007) point out, directly estimating $P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k))$ using relative frequencies (say) is problematic. Indeed, Zens and Ney (2004) showed that the estimation of $P(\hat{e}_k \mid \hat{f}_k)$ using relative frequencies results in the overestimation of the probabilities of long phrases, so smoothing factors in the form of lexical-based features are often used to counteract this bias (Foster et al., 2006). In the case of context informed features, since the context is also taken into account, this estimation problem can only become worse.

To avoid such problems, in this work we use three memory-based classifiers: IG-Tree, IB1 and TRIBL [1] (Daelemans et al., 2007). 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 information gain (IG).

In order to build the set of examples required to train the classifier, we modify the standard phrase-extraction method of (Koehn et al., 2003) to extract the context of the source phrases at the same time as the phrases themselves. Importantly, therefore, the context extraction comes at no extra cost.

We refer the interested reader to (Stroppa et al., 2007) for more details of how Memory-Based Learning (MBL) is used for classification of source examples for use in the log-linear MT framework.

6 Experimental Set-Up

6.1 Features Used

The distribution of target phrases given a source phrase and its contextual information is normalised to estimate $P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k))$. Therefore our expected feature is derived as in (9):

$$\hat{h}_{mbl} = \log P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k)) \quad (9)$$

In addition to the above feature, we derived two more features $\hat{h}_{\text{mod}}$ and $\hat{h}_{best}$ from the posterior probability $P(\hat{e}_k \mid \hat{f}_k)$ and $P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k))$. The feature $\hat{h}_{\text{mod}}$ is defined as in (10):

$$\hat{h}_{\text{mod}} = \log [\alpha\, P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k)) + (1-\alpha)\, P(\hat{e}_k \mid \hat{f}_k)] \quad (10)$$

The interpolation weight α was tuned manually on the devset.

We observed that MBL assigned large weights to more appropriate target phrases rather than less appropriate ones. One interesting observation is that IGTree seems to produce better results on lower α values, while in the case of IB1 and TRIBL, we obtained more mixed results.

[1] An implementation of IGTree, IB1 and TRIBL is freely available as part of the TiMBL software package, which can be downloaded from http://ilk.uvt.nl/timbl.

While the best scores for IB1 and TRIBL were produced at both end of the spectrum, they performed best on higher values of α. Combining these weights, we derived $\hat{h}_{\text{mod}}$. Our final feature $\hat{h}_{best}$ is defined as in (11):

$$\hat{h}_{best} = \begin{cases} 1 \text{ if } \hat{e}_k \text{ maximizes} \\ \qquad P(\hat{e}_k \mid \hat{f}_k, CI(\hat{f}_k)) \\ 0 \text{ otherwise,} \end{cases} \quad (11)$$

We performed three different experiments by integrating these three features $\hat{h}_{mbl}$, $\hat{h}_{\text{mod}}$ and $\hat{h}_{best}$ directly into the log-linear model. In the first experiment E1, the baseline feature log $P(\hat{e}_k \mid \hat{f}_k)$ is directly replaced by $\hat{h}_{\text{mod}}$. In the second experiment (E2), we integrated the $\hat{h}_{mbl}$ feature together with the baseline features, keeping all the features unaffected. In the third experiment (E3), both the features $\hat{h}_{mbl}$ and $\hat{h}_{best}$ are integrated into the model in the same manner. As for the standard phrase-based approach, their weights are optimized using minimum-error-rate training (Och, 2003) for each of the experiments we carried out.

6.2 Pre-Processing

As (Stroppa et al., 2007) point out, PB-SMT decoders such as Pharaoh (Koehn, 2004) or Moses (Koehn, 2007) rely on a static 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 for use with such decoders, the devset and test set that need to be translated is pre-processed. Each word appearing in the test set and devset is assigned a unique id. First we prepare the phrase table using the training data. Then we generate all possible phrases from the development set and test set. These devset and test set phrases are then searched for in the phrase table, and if found, then the phrase along with its contextual information is given to MBL for classification. MBL produces class distributions according to the maximum-match of the features contained in the source phrase. We derive new scores from this class distribution and merge them with the initial information of phrase table to take into account our feature functions ($\hat{h}_{mbl}$, $\hat{h}_{\text{mod}}$ and $\hat{h}_{best}$) in the log-linear model.

In this way we create a dynamic phrase table containing both the standard and the context-informed features. The new phrase table contains the source phrase (represented by the sequence of ids), target phrase and the new score (which varies depending on which experiments (E1, E2 and E3) are being carried out).

A lexicalized re-ordering model was used for all the experiments undertaken. The source phrase in the reordering table is replaced by the sequence of unique ids when the new phrase table is created. By replacing all the words by their ids in the development set, we perform MERT using our new phrase table to optimize the feature weights. In a similar manner, we translate the test set (represented by ids) using our new phrase table.

7 Results and Analysis

Since we intend to use supertags as source side contextual features, we had to choose English as the source language, given that supertag information is currently available for English only.

The experiments were carried out on the English—Chinese data provided by the IWSLT 2006 evaluation campaign, extracted from the Basic Travel Expression Corpus (BTEC). The training, development and test sets contain 40,274, 489 and 486 sentences respectively. This multilingual speech corpus contains sentences similar to those that are usually found in phrase-books for tourists going abroad. It is observed that sentence length of this speech corpus is very small.

Although our main focus was to see the effect on translation quality of incorporating supertags as a source contextual feature, we also carried out experiments with different contextual features (both individually and in collaboration) and with varying windows of context size. The best results obtained from E1, E2 and E3 are reported in the tables.

The results with uniform context size are shown in Table 1. As demonstrated by (Max et al., 2008), it is clear that translation from English can benefit from the addition of source language features, as the inclusion of any type of contextual feature easily improves upon the baseline across all evaluation metrics. Adding source language POS tags adds almost a whole BLEU point (a relative improvement of 4.67%), and further improvements are to be seen when

	BLEU		NIST		WER		PER	
Baseline	*20.56*		*4.67*		*57.82*		*48.99*	
Context length	±1	±2	±1	±2	±1	±2	±1	±2
CCG	21.75	21.52	**4.84**	4.79	**56.28**	**56.95**	48.58	49.10
LTAG	**21.92**	21.34	4.82	4.70	56.63	57.61	**48.43**	49.27
POS	21.52	21.70	4.70	4.76	57.87	57.21	49.62	49.10
Word	21.64	21.59	4.77	4.78	57.15	57.41	49.21	48.37
Word + CCG	21.52	21.53	4.75	4.78	57.21	57.38	48.95	49.45
Word + LTAG	21.64	21.37	4.78	4.79	57.15	57.06	48.89	48.95
Word + POS	21.77	**21.89**	4.78	**4.83**	56.77	56.51	48.58	**48.03**

Table 1: Experiments with uniform context size

Experiment	BLEU	NIST	WER	PER
Baseline	*20.56*	*4.67*	*57.82*	*48.99*
Word±2 + CCG±1	22.01	**4.82**	57.21	**48.63**
Word±2 + LTAG±1	21.38	4.79	57.01	48.89
Word±2 +POS±1	21.61	4.77	**56.78**	48.66
POS±2 + CCG±1	21.08	4.68	58.22	50.05
Word±2+POS±2+CCG±1	21.23	4.72	57.47	49.82
CCG±1 + LTAG±1	21.79	4.74	58.28	49.59
CCG ±1+ LTAG±1[#]	**22.11**	**4.82**	56.95	48.81
Word±1+CCG±1+LTAG±1[#]	21.48	4.79	56.83	48.53
Supertag-Pair±1[#]	21.99	**4.82**	56.83	48.72

Table 2: Experiments with varying context size
([#] Syntactic features of focus phrase are ignored)

	Experiment	BLEU	NIST	WER	PER
	Baseline	*20.56*	*4.67*	*57.82*	*48.99*
IB1	CCG±1	22.08	4.83	57.30	48.63
IB1	LTAG±1	22.06	4.75	58.05	49.04
IB1	CCG±1+LTAG±1[#]	21.72	4.76	58.48	49.18
IB1	Supertag-Pair±1[#]	22.03	4.79	57.35	49.15
TRIBL	CCG±1	**22.18**	**4.85**	**56.31**	**48.55**
TRIBL	LTAG±1	21.39	4.78	56.83	48.72
TRIBL	CCG±1+LTAG±1[#]	22.00	4.75	58.16	49.59
TRIBL	Supertag-Pair±1[#]	22.13	4.80	57.24	48.92

Table 3: Experiments with IB1 and TRIBL

neighbouring words (5.25% relative increase), CCG supertags (5.79%) and LTAG supertags (6.61%) are used.

Interestingly, with respect to BLEU score, for all bar POS tags and the combination of Word+POS, the best scores are observed when a context window of ±1 words is seen. When ±2 words are used, CCG supertags when used as an individual feature produce the best NIST, WER and PER scores (though these scores are slightly worse than when a context window of ±1 words is used).

When combinations of two features were applied, the Word+POS combination improved BLEU, NIST and PER scores on a ±2 word context window, but no combination improved over the LTAG individual feature when used on a ±1 word context window. Interestingly, when used together with the neighbouring words as a feature, the supertags could not improve over the Words feature, and in most cases caused system performance to deteriorate.

Since LTAG±1 and POS±2 produced the best BLEU scores when used as individual features, we were encouraged to try out combinations of features with varying context sizes. The results can be seen in Table 2. This time, adding CCG supertags to the neighbouring words caused system performance to improve to 22.01 BLEU score, 1.45 points (or 7.05% relative improvement) over the PB-SMT baseline. Encouragingly, the best performance of all was seen when both supertag features were used in combination. Here a BLEU score of 22.11 (7.54% relative improvement compared to the baseline) was obtained for CCG±1+LTAG±1, when ignoring the syntactic feature information of the focus phrase. We also carried out the best performing experiments on IB1 and TRIBL classifiers, with the results shown in Table3. The differences we see between IGTree, TRIBL, and IB1 are generally small and somewhat unpredictable. When considered as a single concatenated feature, the supertag-pair (LTAG, CCG) performed best on TRIBL. When the supertags are used as a standalone feature, IB1 produced the best score on LTAG (7.3% relative improvement), and TRIBL on CCG (7.88% relatively better). Among the three classifiers, however, the IGTree score remains the best on CCG±1+LTAG±1.

8 Conclusion and Future Work

In this paper, we have successfully incorporated supertags as a new feature into a state-of-the-art log-linear phrase-based SMT system that takes into account the contextual information of the source phrases. In addition, we have demonstrated that both neighbouring words and the POS tags of those words can improve translation quality significantly over the baseline system for English-to-Chinese.

Our best result of 1.62 BLEU points improvement over the baseline, a 7.88% relative increase in performance, came about on CCG alone. Most encouragingly, supertags produced good results consistently.

Following the work of (Hassan et al., 2006, 2007, 2008; Hassan, 2009), our ultimate aim is to develop a fully supertagged PB-SMT system, with supertags deployed as source language context (as here), as well as in the target language model and the target side of the t-table. We have been made aware that a German version of the CCGBank may be available, but so far we have been unable to verify this. We will continue to pursue this line of investigation, with a view to benefiting from clear the advantages that supertags bring to bear in each phase of the translation process.

Other lines of future work include (i) a manual evaluation of the output sentences, to try to identify the exact role that supertags are playing when used as source language contextual information; (ii) an investigation as to why system performance tends to deteriorate when pairs of features are used, and where one of those pairs is a supertag sequence; and (iii) an investigation as to why a context window of ±1 words seems to work better than larger windows.

Acknowledgements

We would like to thank our colleague Hany Hassan for his input on the use of supertags. We are grateful to SFI (http://www.sfi.ie) for generously sponsoring this research under grants 05/IN/1732 and 07/CE/I1142.

References

Bangalore, Srinivas and Aravind K. Joshi. 1999. Supertagging: An Approach to Almost Parsing. *Computational Linguistics* **25**(2):237–265.

Berger, Adam, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, **22**(1):39–68.

Carpuat, Marine, and Dekai Wu. 2005. Word sense disambiguation vs. statistical machine translation. *ACL-2005, 43rd Annual meeting of the Association for Computational Linguistics*, Ann Arbor, MI, 387—394.

Carpuat, Marine, and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. *EMNLP-CoNLL-2007, Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic, 61—72.

Chan, Y. S., H. T. Ng., and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 33—40.

Chen, J., Srinivas Bangalore and K. Vijay-Shankar. 2006. Automated Extraction of Tree-Adjoining Grammars from Treebanks. *Natural Language Engineering* **12**(3):251—299.

Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics* 33(2): 202—228.

Clark, Steven and James Curran. 2004. The Importance of Supertagging for Wide-Coverage CCG Parsing. *Coling-2004. 20th International Conference on Computational Linguistics*, Geneva, Switzerland, 282–288.

Cowan, Brooke, Ivona Kučerová, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. *EMNLP-2006: Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 232—241.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot and Antal van den Bosch, A. 2007. *TiMBL: Tilburg Memory Based Learner, version 6.1, Reference Guide*. ILK Research Group Technical Report Series no. 07-07.

Foster, George, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. *EMNLP-2006: Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 53—61.

García-Varea,, Ismael, F. J. Och, H. Ney, and F. Casacuberta. 2001. Refined lexicon models for statistical machine translation using a maximum entropy approach. *ACL-2001, 39th Annual Meeting of the Association for Computational Linguistics and 10th Meeting of the European Chapter*, Toulouse, France, 204—211.

Giménez, Jesús, and Lluís Màrquez. 2007. Context-aware discriminative phrase selection for statistical machine translation. *ACL 2007: proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, 159—166.

Gimpel, Kevin and Noah A. Smith. 2008. Rich source-side context for statistical machine translation. *ACL-08: HLT. Third Workshop on Statistical Machine Translation*, Columbus, OH, pp.9—17.

Hassan, Hany. 2009. *Lexical Syntax for Statistical Machine Translation.* Ph.D Thesis, Dublin City University, Dublin, Ireland.

Hassan, Hany, Mary Hearne, Khalil Sima'an, and Andy Way. 2006. Syntactic Phrase-Based Statistical Machine Translation. *IEEE 2006 Workshop on Spoken Language Translation*, Palm Beach, Aruba.

Hassan, Hany, Yanjun Ma, and Andy Way. 2007. MaTrEx: the DCU Machine Translation System for IWSLT 2007. In *Proceedings of the International Workshop on Spoken Language Technologies*, Trento, Italy, pp.69—75.

Hassan, Hany, Khalil Sima'an, and Andy Way. 2007. Supertagged phrase-based statistical machine translation. *ACL-2007. 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 288—295.

Hassan, Hany, Khalil Sima'an, and Andy Way. 2008. Syntactically Lexicalized Phrase-Based SMT. *IEEE Transactions on Audio, Speech and Language Processing* **6**(7):1260—1273.

He, Zhongjun, Qu Liu and Shouxu Lin. 2008. Improving statistical machine translation using lexicalized rule selection. In *Coling 2008: Proceedings of the Conference*, Manchester, UK, pp.321—328.

Hockenmaier, Julia. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, UK.

Ittycheriah, Abe and Salim Roukos. 2007. Direct Translation Model 2. *NAACL-HLT-2007, Human Language Technology: the conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, NY, pp.57—64.

Joshi, Aravind and Yves Schabes. 1992. Tree Adjoining Grammars and Lexicalized Grammars. In M. Nivat and A. Podelski (eds.) *Tree Automata and Languages*, Amsterdam, The Netherlands: North-Holland, pp.409–431.

Koehn, Philipp. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Machine translation: from real users to research: 6th conference of the Association for Machine Translation in the Americas, AMTA 2004*, Berlin: Springer Verlag, 2004, 115—124.

Koehn, Philipp, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. *ACL 2007: proceedings of demo and poster sessions*, Prague, Czech Republic, 177-180.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. *HLT-NAACL 2003, conference combining Human Language Technology conference series and the North American Chapter of the Association for Computational Linguistics conference series*, Edmonton, Canada, 48-54.

Liang, Percy, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. *Coling-ACL 2006: 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, 761—768.

Max, A., R. Makhloufi and P. Langlais. 2008. Explorations in Using Grammatical Dependencies for Contextual Phrase Translation Disambiguation. In *Proceedings of the 12th EAMT Conference*, Hamburg, Germany, pp.112—117.

Och, Franz. 2003. Minimum error rate training in statistical machine translation. *41st Annual meeting of the Association for Computational Linguistics*, Sapporo, Japan, 160–167.

Och, Franz, and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. *40th Annual meeting of the Association for Computational Linguistics*, Philadelphia, PA, 295–302.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. *40th Annual meeting of the Association for Computational Linguistics*, Philadelphia, PA, 311–318.

Steedman, Mark. 2000. *The Syntactic Process.* MIT Press: Cambridge, MA.

Stroppa, Nicolas, Antal van den Bosch and Andy Way. 2007. Exploiting Source Similarity for SMT using Context-Informed Features. *TMI-2007, 11th Conference on Theoretical and Methodological Issues in Machine Translation*, Skövde, Sweden, 231—240.

Vickrey, David, Luke Biewald, Marc Teyssier and Daphne Koller. 2005. Word-sense disambiguation for machine translation. *HLT-EMNLP-2005, Human Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, 771—778.

Zens, Richard and Hermann Ney. 2004. Improvements in phrase-based statistical machine translation. *HLT/NAACL 2004, Human Language Technology conference/North American Chapter of the Association for Computational Linguistics annual meeting*, Boston, MA, 257–264.

On LM Heuristics for the Cube Growing Algorithm

David Vilar and Hermann Ney
Lehrstuhl für Informatik 6
RWTH Aachen University
52056 Aachen, Germany
{vilar,ney}@informatik.rwth-aachen.de

Abstract

Current approaches to statistical machine translation try to incorporate more structure into the translation process by including explicit syntactic information in form of a formal grammar (with a possible, but not necessary, correspondence to a linguistic motivated grammar). These more structured models incur into an increased generation cost, and efficient algorithms must be developed. In this paper we concentrate on the *cube growing* algorithm, a lazy version of the cube grow algorithm. The efficiency of this algorithm depends on a heuristic for language model computation, which is only scarcely discussed in the original paper. In this paper we investigate the effect of this heuristic on translation performance and efficiency and propose a new heuristic which efficiently decreases memory requirements and computation time, while maintaining translation performance.

1 Introduction

In the last decade, the phrase-based approach to machine translation has been the de-facto standard for statistical machine translation (SMT) systems. The main reason was that it offered a great improvement in translation quality over its predecessors, the single-word based models. The model itself is also relatively simple and allows for efficient generation algorithms like for example beam search (see e.g. (Koehn, 2004)). This allowed the approach to scale to bigger tasks and it is still one of the most widely used models nowadays. The current trend in SMT, however, is to bring more information in the form of grammatical structures. There are mainly two possibilities, in the form of linguistically motivated grammars (e.g. (Marcu et al., 2006)) or just formal grammars, which do not need to have a linguistic equivalent (e.g. (Chiang, 2005)).

These more expressive models have associated a more difficult search problem, which normally involves a parsing process (usually a variation of the CYK algorithm) while incorporating the translation information. The inclusion of language model (LM) information replicates nodes in the parsing tree, which increases the cost of the generation process. And not to be underestimated, the number of rules the system has to deal with can be of one order of magnitude bigger than the standard phrase-based approach, depending on the model[1].

Therefore, new, efficient algorithms for translation with these richer models had to be developed. In this paper we will concentrate on the cube growing algorithm (Huang and Chiang, 2007), a lazy version of the cube pruning algorithm (Chiang, 2007). These algorithms represent the search space as an hypergraph and add language model scores as necessary.

The most time-consuming operation in the translation process is the LM score computation, especially when huge LMs are used. The cube growing algorithm follows an on-demand computation strategy and tries to minimize the number of LM scores that need to be computed. In order to minimize the number of search errors, while still maintaining computational efficiency, the algorithm depends on an (efficient) heuristic for these LM costs, which is only scarcely dis-

[1] Note that most of these models do not discard the (majority of) standard phrases, instead they add new rules to the phrase inventory.

Proceedings of the 13th Annual Conference of the EAMT, pages 242–249,
Barcelona, May 2009

cussed in the original paper.

In this work we investigate the originally proposed heuristic in terms of translation quality and computational cost and propose a new heuristic, which maintains translation performance while reducing memory requirements at no computation time expense. The main idea is to cluster the words in the target language into a reduced number of classes and to compute an optimistic LM score on these classes, a concept similar to the one presented in (Petrov et al., 2008). We focus our attention on the hierarchical phrase-based model proposed in (Chiang, 2007), but our findings may as well be applicable to other translation models.

This paper is structured as follows. Section 2 reviews the hierarchical phrase-based approach to SMT and Section 3 the cube growing algorithm. In Section 4 we present the requirements for the heuristics to be used in this algorithm. Section 5 presents our new heuristic. Experimental results are presented in Section 6 and discussed in Section 7. The paper concludes in Section 8.

2 The Hierarchical Phrase Based Approach

The hierarchical phrase-based approach can be considered as an extension of the standard phrase-based model. In this model we allow the phrases to have "gaps", i.e. we allow non-contiguous parts of the source sentence to be translated into possibly non-contiguous parts of the target sentence. The model can be formalized as a synchronous context-free grammar (Chiang, 2007). The bilingual rules are of the form

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle \,, \tag{1}$$

where X is a non-terminal, γ and α are strings of terminals and non-terminals, and $\sim$ is a one-to-one correspondence between the non-terminals of α and γ.

Two examples of this kind of rules for the German-to-English translation direction are

$$X \rightarrow \langle \text{ich habe } X^{\sim 0} \text{ gesehen}, \text{I have seen } X^{\sim 0} \rangle$$
$$X \rightarrow \langle \text{im } X^{\sim 0} \text{ zu } X^{\sim 1}, \text{in order to } X^{\sim 1} X^{\sim 0} \rangle$$

where the indices in the non-terminals represent the correspondence between source and target "gaps". This model has the additional advantage that reordering is integrated as part of the model itself, as can be seen in the above examples.

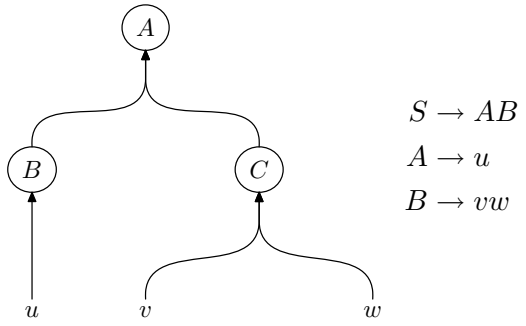


Figure 1: Example of an hypergraph corresponding to a grammar derivation. The hypergraph corresponds to the derivation of the string uvw using the grammar shown on the right.

The first step in the hierarchical phrase extraction is the same as for the phrased-based model. Having a set of initial phrases, we search for phrases which contain other smaller sub-phrases and produce a new phrase with gaps. In our system, we restricted the number of non-terminals for each hierarchical phrase to a maximum of two, which were also not allowed to be adjacent in the source side, and the gaps were allowed to have a maximum size of 10 words. The scores of the phrases are computed as relative frequencies.

3 The Cube Growing Algorithm

Starting point for the cube growing algorithm is a representation of the parsing space by means of an hypergraph. An hypergraph is an extension of the standard graph concept, where each (hyper-)node can have more than one predecessor. The hyperedges have thus an arity, which indicates the number of predecessors of the goal node. An example visualization of a simple derivation is shown in Figure 3. In our case the hypergraph is generated applying the CYK+ algorithm (Chappelier and Rajman, 1998) on the source sentence.

In the following, an informal description of the cube growing algorithm will be given, stressing the usage of heuristic function for the LM score computation. For a full detailed description, the reader is advised to consult (Huang and Chiang, 2007). In our description we will use the terms derivation and translation interchangeably. Remember that the hierarchical model is represented as a parallel grammar. Therefore, given a derivation of the source sentence, we can construct a corresponding derivation in the target language and thus obtain a translation. It is true that the same translation

(considered only at the word level) may arise from different source derivations. However, a strict distinction is not necessary in this context and would only clutter the text.

We will consider cost minimization in the following exposition, i.e. the best derivation is the one with a minimum cost, and costs are combined by adding them. The main procedure of the cube growing algorithm finds the n-th best derivation of a given node in an hypergraph. In order to do this, it recursively calls itself on the predecessor nodes, computing the necessary subderivations on demand. E.g. assume that the 10-th best derivation of an hypernode is formed by combining the 2nd-best and the 4-th best derivation of two predecessor hypernodes. In this case, only those 2 and 4 derivations of the predecessor nodes would have been computed[2].

If no LM score is taken into account, this computation can be carried out in an exact way. The different translation alternatives for one hyperedge can be sorted according to their costs. The derivations in the hypernodes with no predecessors (purely lexical rules), can thus be generated in a monotonic way. This allows, with a proper combination strategy, to generate the derivations in every hypernode in a cost-increasing order.

When including the LM score, however, the situation is different. The costs of the derivation are no longer simply the sum of the corresponding derivations in the predecessor hypernodes plus the cost of the grammar rule. Now we have to add the cost of the language model computed on the associated target language parts. The language model score is called a *combination cost*, as it is a cost that affects the combination of hypernodes. This score is costly to compute and dependent on all elements participating in the combination (predecessor hypernodes and translation rule). The effect of this is that the sorting strategy referred to above cannot longer guarantee that the generation of the derivations in an hypernode will proceed in a monotonic order.

In order to overcome this problem, we store the generated derivations in an intermediate buffer and we will only extract them from this buffer when we are confident that no better derivation can be found for the given hypernode. In the original paper it is shown that, if one can define an heuristic for the LM costs along an hyperedge, this technique will generate an exact n-best list of derivations.

The original paper proposes to compute an n-best list of translations without taking into account the LM scores, a so-called "-LM" parse (possibly taking into account recombination of hypotheses). Afterwards they compute the LM scores of these n-best derivations, and use these scores as the heuristic for the hyperedges involved in these derivations. The motivation behind this approach is that in the -LM pass, we will compute a hopefully representative portion of the needed derivations and thus, the best of these scores should act as an heuristic for the hyperedge. Note however, that there is no guarantee that the explored space will be big enough. If when taking the LM into account (the "+LM" parse) we need the heuristic for a hyperedge which was not computed in the -LM pass, we just take the LM score of the first-best derivation for this hyperedge.

Note that there is also an additional parameter for controlling the computational cost by limiting the number of derivation to be taken into account, i.e. the size of the intermediate buffer. This parameter can also have an effect when counteracting ill-formed heuristics.

4 Heuristic requirements

In order for the algorithm not to produce search errors, the heuristic must be optimistic (also called "acceptable"), that is, the costs given by the heuristic must be less than the actual cost. If this can be guaranteed, it can be shown that the search algorithm does not produce any search errors.

Another key issue for practical application is the necessity that the heuristic computation must be efficient. If too much time is spent on computing the heuristic, the gains of the lazy evaluation can be overcome by this computation time. In the extreme case, we could compute the LM cost of all possible combination at each hypernode, which will lead to an optimal heuristic. Of course this computation would be much more costly than the actual search using the cube growing algorithm.

In the case of the -LM heuristic, we can not guarantee its acceptability, as we cannot show that the hyperedges used in the -LM n-best computation will be reused in the +LM parse. In fact, the translations produced without language model differ much from the ones when the language model is taken into account, therefore it is not clear the

[2]By comparison, the cube pruning algorithm computes a fixed number of derivations at each hypernode.

adequacy of this heuristic. The efficiency can be controlled by varying the size of the n-best list, however small values of n can increase the risk of inappropriate heuristic values.

5 Coarse LM Heuristic

In this section we propose a new heuristic for the score computation of the members of the intermediate buffer. We first recall that, given an n-gram language model, the score of a word w given its context h (also called history) is given by the expression (Kneser and Ney, 1995)

$$p(w|h) = \begin{cases} \alpha(w|h) & \text{if } N(h,w) > 0 \\ \gamma(h)\alpha(w|\overline{h}) & \text{if } N(h,w) = 0 \end{cases} \quad (2)$$

where $N(h,w)$ corresponds to the word-history count in the training corpus, $\alpha(w|h)$ is the (discounted) relative frequency of the word-history pair, $\gamma(h)$ is a back-off weight, which also ensures a proper normalization of the probability distribution and $\overline{h}$ is a generalized history, that is, h with the last word dropped.

Now assume we have a mapping $\mathcal{C}$ from our target vocabulary V into a set of classes K, with $|K| \ll |V|$

$$\begin{aligned} \mathcal{C} : V &\to K \\ w &\mapsto \mathcal{C}_w \end{aligned} \quad (3)$$

We can extend the mapping to a sequence of words w_1^N just by concatenating the mappings of the individual words, i.e. $\mathcal{C}_{w_1^N} = \mathcal{C}_{w_1} \dots \mathcal{C}_{w_N}$.

Given this mapping we now define our heuristic by taking the maximum LM probability associated with the words that get mapped to the same class. More formally, define the following functions corresponding to the quantities α and γ of equation 2

$$\alpha_{\mathcal{H}}(w|h) = \max_{\substack{w':\mathcal{C}_{w'}=\mathcal{C}_w \\ h':\mathcal{C}_{h'}=\mathcal{C}_h}} \left\{\alpha(w'|h')\right\} \quad (4)$$

$$\gamma_{\mathcal{H}}(h) = \max_{h':\mathcal{C}_{h'}=\mathcal{C}_h} \left\{\gamma(h')\right\} \quad (5)$$

and the resulting heuristic

$$\mathcal{H}(w|h) = \begin{cases} \alpha_{\mathcal{H}}(w|h) & \text{if } N(\mathcal{C}_w|\mathcal{C}_h) > 0 \\ \gamma_{\mathcal{H}}(h)\alpha_{\mathcal{H}}(w|\overline{h}) & \text{if } N(\mathcal{C}_w|\mathcal{C}_h) = 0 \end{cases} \quad (6)$$

The parameters of this heuristic function can be computed offline before the actual translation process and are stored in ARPA-format, like a normal LM. This allows the reuse of the existing code for handling language models.

Note that $\mathcal{H}(w|h)$ does not define a probability distribution any more, as it is not normalized. This poses no problem, as we are looking for an upper bound of the language model probabilities, and these do not need to form a probability distribution themselves.

This heuristic value is computed for the derivations as they are being produced, and it gets updated in the corresponding hyperedge. The motivation for this heuristic is that the expected similarity of the words which can be produced by the translation rules associated with an hyperedge and the contexts in this hyperedge can be captured with the given classes, and thus this optimistic language model score is able to predict future LM scores.

One could also think of a, at least at first glance, more straightforward approach. Given the mapping of words into classes, we could compute the mapping of the data used for training the language model, and then train a new language model on this data. This approach, however, has a big drawback for the usage as an heuristic. If a new language model is trained, the probabilities associated with it are in a completely different range, due to the reduced vocabulary size. Therefore the newly trained language model does not give enough information about the original language model.

5.1 Acceptability

Is this heuristic optimistic (and thus acceptable)? Taking into account the derivations for which we compute the heuristic, in most of the cases it is. This is because we take the maximum of every term involved in Equation 2. Note however, that the conditions in the case distinction have changed. In particular we move from testing the presence of a word-history pair to the presence of the corresponding classes. As the classes are more general than the words it can be the case that for some combination we use the event-seen case (first line in the case distinction of Equations 2 and 6) instead of the backoff case used when considering the words themselves. In practice, the probability of the event-seen case is expected to be higher, but we can not guarantee it.

Another source of discrepancy arises from the term $\gamma(h)$ (and the corresponding $\gamma_{\mathcal{H}}(h)$) and unseen histories h. Again, it can happen that in considering $\mathcal{C}_h$ we shift from an unseen to a seen event. Depending on the definition of the γ function this can have issues on the acceptability of the heuristic

function. In our concrete case, we train our models using Kneser-Ney smoothing (Kneser and Ney, 1995) and use the SRI toolkit (Stolcke, 2002) for our implementation. Under this conditions, for unseen histories, $\gamma(h) = 1$ (or gets a cost of 0, in the negative log-probability space). That means that when C_h has been seen, our heuristic will again not be acceptable. This, however, does not seem to have a big negative effect on the results.

The generalization on other hypotheses along the same hyperedge cannot be guaranteed, but experiments on this respect are presented on Section 7.

With respect to efficiency, this heuristic introduces a new language model into the translation process. However, the size of this language model is quite small, especially when compared with the full language model used in search, and thus the overhead of the additional LM computations is small. On the other side, when compared with the original heuristic, we eliminate the need of the -LM pass altogether.

5.2 Choosing the Classes

There is still the open question of how to choose the word-to-class mapping $\mathcal{C}$. In our case we investigated two alternatives. The first one is to use automatically generated classes. We used the `mkcls` tool (Och, 1999), which uses a maximum likelihood approach on a corpus by using a class bigram decomposition. This tool is widely used as part of the preprocessing steps when training statistical alignments using the GIZA++ tool (Och and Ney, 2003). This criterion seems to be adequate for our task, as both the words themselves and the context are taken into account.

Another possibility would be to use Part-of-Speech tags as word classes. The tagging itself can, however, be an expensive process, involving a new search in itself. We applied a simplifying assumption, in which we remove the ambiguity of the tagging. We applied a full POS-tagger (Brants, 2000) to the training corpora and then we simply selected the most frequent POS tag for each word. In this way we defined our mapping $\mathcal{C}$.

6 Experimental Results

Experiments are reported using the 2008 WMT evaluation data (Callison-Burch et al., 2008), for the German-to-English translation direction. This corpus consists of the speeches held in the plenary session of the European Parliament. The test data was the in-domain data used in the evaluation. The statistics of the corpus can be seen in Table 1.

Figure 2 shows the results for the -LM heuristic[3]. The BLEU score is shown in Figure 2(a). The best results are achieved with a -LM n-best size of 200. The difference in performance, however is not too big and nearly optimal results can already be achieved with a -LM n-best size of 50. When looking into the computational resources the difference, however, becomes critical. Figure 2(b) shows the memory usage dependent on the -LM n-best size. We can see that the memory requirements grow nearly linearly with the size of the n-best list (which is to be expected). The memory requirements using a -LM 50-best list is around 1.6GB. When using the 200-best list for optimal performance the memory requirements grow up to 6.5GB. For n-best sizes greater then 400, the memory requirements become prohibitive for the majority of current computers.

Computation time requirements are shown in Figure 2(c), as the average time needed for translating a sentence. The time requirements also grow with increasing -LM n-best size, but they stay quite reasonable, with a maximum of 6.5s per sentence. For optimum performance (200-best list), 5.2s per sentence are needed, for a 50-best heuristic, 4.3s. All time measures were taken on machines equipped with Quad-Core AMD Opteron processors, with a clock speed of 2.2GHz.

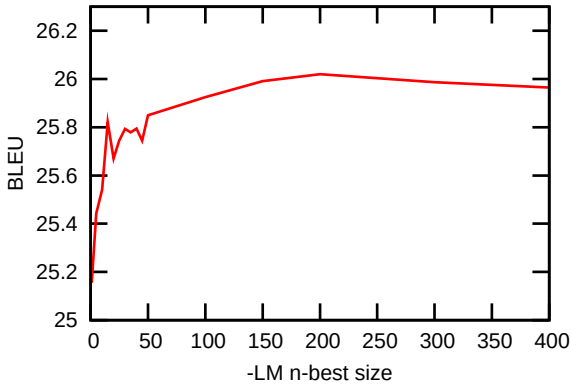
The results for the coarse LM heuristic are show in Figure 3. It can be seen that the performance of the system using this heuristic is comparable or even slightly superior in the best case, however only marginally so. The behaviour of this heuristic is somewhat more erratic than in -LM case. Memory requirements are shown in Figure 3(b). The memory requirements using the coarse LM heuristic are much lower than when using the -LM heuristic (note the different scale on the y-axis between Figures 2(b) and 3(b)), and they get less as the number of classes increases.

Time requirements are shown in Figure 3(c) and are in general lower than for the case of -LM heuristics, except for very small values of n, where the translation performance suffers severely. The time requirements also show an erratic behaviour. However, different workloads of the ma-

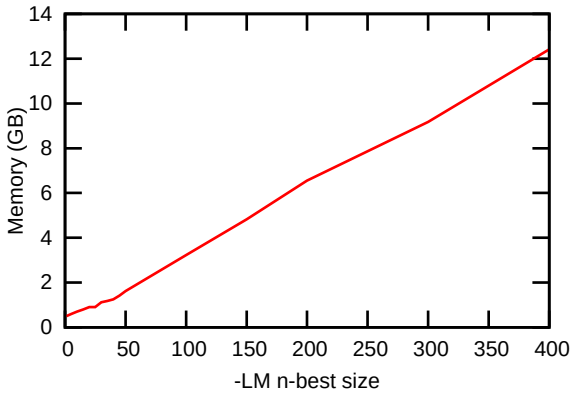
[3]Note that in our implementation we used hypothesis recombination also in the -LM pass

	Training set		Test set	
	German	English	German	English
Sentences	1,266,520		2,000	
Words	33 404 503	35 259 758	56 624	60 185
Distinct words	301 006	96 802	8 844	6 050

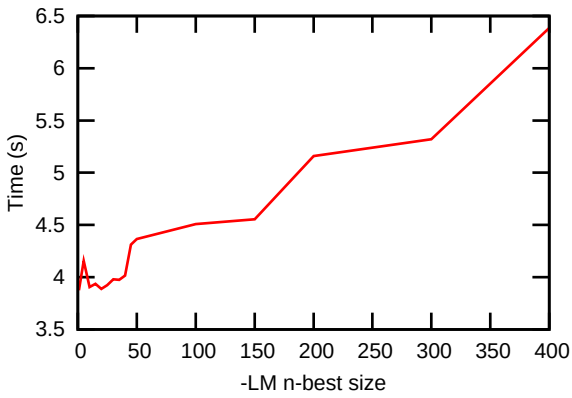
Table 1: Corpora statistics



(a) BLEU score

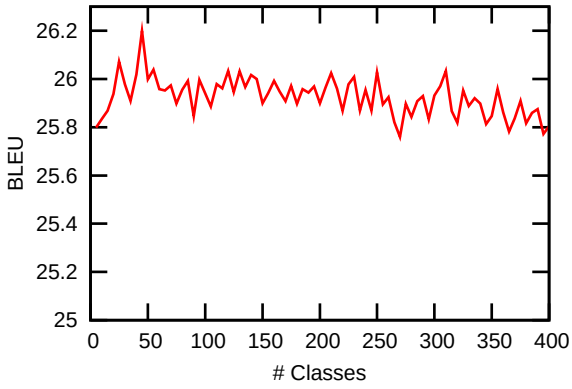


(b) Memory

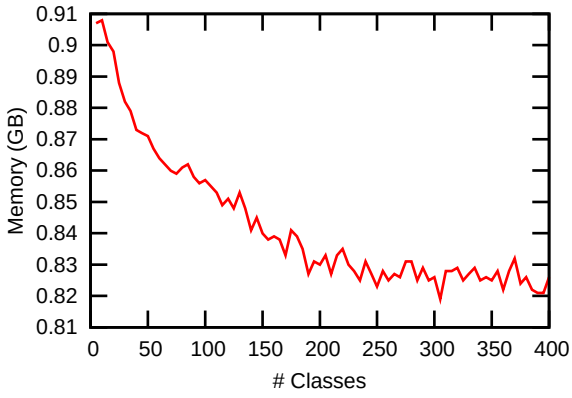


(c) Time per sentence

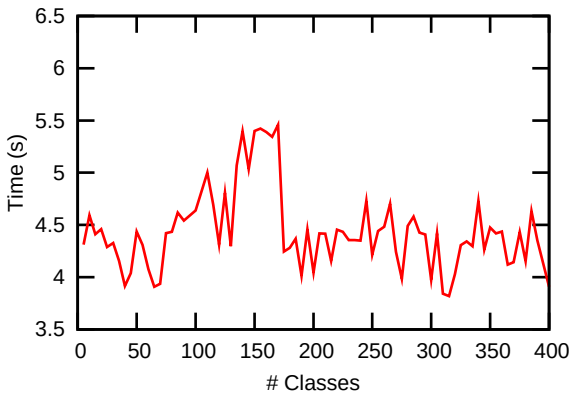
Figure 2: Results using the -LM heuristic



(a) BLEU score



(b) Memory



(c) Time per sentence

Figure 3: Results using the coarse LM heuristic

chines at experimentation time probably had a non-negligible effect on these measurements.

Using POS as classes does not seem to improve performance. It achieves a BLEU score of 25.9%, and the memory and time requirements are comparable with those of the equivalent number of automatic classes (we work with 41 POS classes).

7 Discussion

The behaviour of the -LM heuristic was expected. The increase in memory and time requirements is due to the increase effort in generating the -LM n-best lists. This does not imply an increase in translation quality, as, probably, the new hyperedges that get considered in the heuristic computation do not get used in the actual translation process.

Figure 4 shows how many times the -LM heuristic was not acceptable, computed for the first 100 sentences of the test corpus. As expected, this number decreases as the size of the n-best list increases, but starting from a value of around 100 the rate of decrease of failed heuristics is much lower. This explains the behaviour of the BLEU score shown in Figure 2(a).

The coarse-LM heuristic already achieves a good performance even for a small number of classes. This heuristic is able to simplify the LM computation scores and guide the parsing process in an efficient manner. This is consistent with the findings of (Petrov et al., 2008), albeit in a slightly different context (Petrov et al. used the coarse LM for pruning purposes).

This observation can be confirmed in Figure 5, where the coarse heuristic produces much less heuristic failures as the -LM heuristic. Somewhat counter-intuitively, however, the number of LM heuristic fails increases with the number of classes. This can be explained by the fact, that, in spite of the chances of incurring into one of the failure cases exposed in Section 5.1 grow when considering a small amount of classes, the maximum probability induced by the mapping C also is greater as the number of classes diminishes. This can be clearly seen by considering only one class. In spite of certainly incurring in one of the "fail" cases, we certainly will get an optimistic estimation of the LM score. In this way, a small number of classes does not provide so good information to accurately discriminate between the candidate derivations, which explains the higher cost with a small number of classes. In this case, the absolute bound on the number of candidates pointed to at the end of Section 3 has a higher probability of coming into effect.

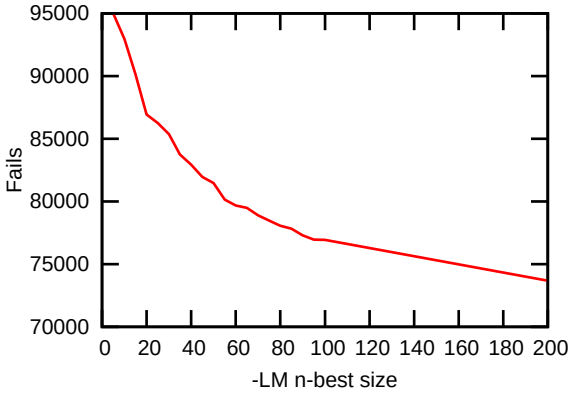


Figure 4: -LM heuristic fails.

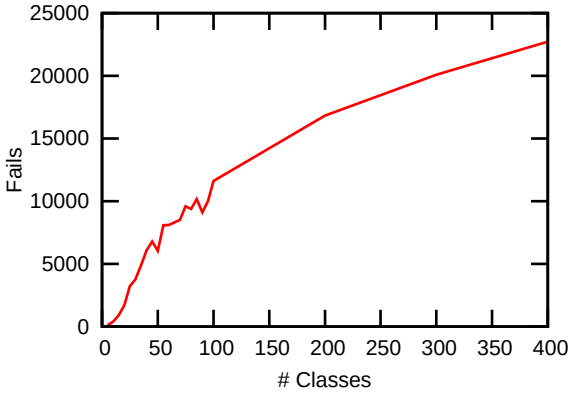


Figure 5: Coarse LM heuristic fails.

An increase in the number of classes reduces the memory and time requirements, which is an indication that the search effort gets more focused. This is not reflected in an increase in BLEU score. The behaviour of the BLEU curve is however somewhat erratic and, nevertheless stays on a range of 0.2%, so no clear conclusion can be drawn from that. Of course, an increase in the number of classes also implies an increase in the lookup time (but with sizes up to 400, as we have done in this paper, this is negligible). This also involves an increase in (offline) computation time for clustering the words. In our experiments we selected the maximum number of classes that we could compute in 24 hours[4].

[4]However on some less powerful computers as the ones the translation experiments were carried on

8 Conclusions

In this paper we have studied the language model heuristic proposed by (Huang and Chiang, 2007) where they describe the cube growing algorithm. We have analysed the performance and efficiency when varying the size of the n-best list required for the heuristic computation. We have proposed a new heuristic, based on taking the maximum of the LM scores which achieves the same (or marginally better) performance but using significant less memory and with improvements in running time.

We just tried two approaches to word clustering, the automatic one implemented by the well-known tool `mkcls` and the one based on POS classes. Although no big difference in performance or efficiency could be found between these two word clusterings, perhaps a smarter and more task-directed clustering criterion can further improve the results. Specially, a bilingual word clustering algorithm, where both the source and target language words are taken into account, will probably provide better performance, as the similarity between the words that may occur as translations along an hyperedge may be better modelled.

Acknowledgements

This work was realised as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. 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 (DARPA).

References

Brants, Thorsten. 2000. Tnt – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP)*, pages 224–231, Seattle, WA.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio, June. Association for Computational Linguistics.

Chappelier, JC and M. Rajman. 1998. A generalized CYK algorithm for parsing stochastic CFG. In *First Workshop on Tabulation in Parsing and Deduction (TAPD98)*, pages 133–137.

Chiang, D. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. *Ann Arbor*, 100.

Chiang, D. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228.

Huang, L. and D. Chiang. 2007. Forest Rescoring: Faster Decoding with Integrated Language Models. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 144.

Kneser, R. and H. Ney. 1995. Improved backing-off for M-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1.

Koehn, P. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Washington DC*.

Marcu, D., W. Wang, A. Echihabi, and K. Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. *Proceedings of EMNLP*, pages 44–52.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, F.J. 1999. An efficient method for determining bilingual word classes. In *EACL99: Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics*, pages 71–76.

Petrov, Slav, Aria Haghighi, and Dan Klein. 2008. Coarse-to-fine syntactic machine translation using language projections. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 108–116, Honolulu, Hawaii, October. Association for Computational Linguistics.

Stolcke, A. 2002. SRILM-an Extensible Language Modeling Toolkit. In *Seventh International Conference on Spoken Language Processing*. ISCA.

Tuning Syntactically Enhanced Word Alignment for Statistical Machine Translation

Yanjun Ma[†] **Patrik Lambert**[‡] **Andy Way**[†‡]
[†]National Centre for Language Technology
[‡]Centre for Next Generation for Localisation
School of Computing, Dublin City University
Dublin 9, Ireland
{yma,plambert,away@computing.dcu.ie}

Abstract

We introduce a syntactically enhanced word alignment model that is more flexible than state-of-the-art generative word alignment models and can be tuned according to different end tasks. First of all, this model takes the advantages of both unsupervised and supervised word alignment approaches by obtaining anchor alignments from unsupervised generative models and seeding the anchor alignments into a supervised discriminative model. Second, this model offers the flexibility of tuning the alignment according to different optimisation criteria. Our experiments show that using our word alignment in a Phrase-Based Statistical Machine Translation system yields a 5.38% relative increase on IWSLT 2007 task in terms of BLEU score.

1 Introduction

Word alignment, which can be defined as a problem of determing word-level correspondences given a parallel corpus of aligned sentences, is a fundamental component in Phrase-Based Statistical Machine Translation (PB-SMT). The dominant approach to word alignment are generative models, including IBM models (Brown et al., 1993) and HMM models (Vogel et al., 1996; Deng and Byrne, 2006). While generative models trained in an unsupervised manner can produce high-quality alignments given a reasonable amount of training data, it is difficult to incorporate richer features into such models. On the other hand, discriminative models are more flexible to incorporate arbitrary features. However, these models need a certain amount of annotated word alignment data, which is often subject to criticism since the annotation of word alignment is a highly subjective task. Moreover, parameters optimised on manually annotated data are not necessarily optimal for MT tasks. Recent research attempts to combine the merits of both generative and discriminative models (Fraser and Marcu, 2007), or to tune a discriminative model according to MT metrics (Lambert et al., 2007).

In this paper, we introduce a simple yet flexible framework for word alignment. To take the advantage of the strength of generative models, we use these models to obtain a set of anchor alignments. We then incorporate syntactic features induced by the anchor alignments into a discriminative word alignment model. The syntactic features we used are syntactic dependencies. This decision is motivated by the fact that if words tend to be dependent on each other, so does the alignment (Ma et al., 2008). If we can first obtain a set of reliable anchor links, we could take advantage of the syntactic dependencies relating unaligned words to aligned anchor words to expand the alignment. Figure 1 gives an illustrative example. Note that the link (c_2, e_4) can be easily identified, but the link involving the fourth Chinese word (a function word denoting 'time') (c_4, e_4) is hard. In such cases, we can make use of the dependency relationship ('tclause') between c_2 and c_4 to help the alignment process.

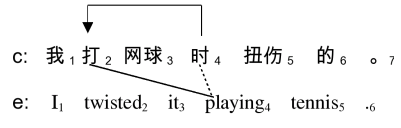


Figure 1: Dependencies for word alignment

Proceedings of the 13th Annual Conference of the EAMT, pages 250–257,
Barcelona, May 2009

Our experiments show that using our word alignment approach in a PB-SMT system can significantly improve the system over a strong baseline. The experiments also show that syntax is beneficial in word alignment. Given that the intrinsic quality of word alignment measured using F-score does not correlate well with PB-SMT performance measured using BLEU, we conducted experiments that can directly optimise the word alignments according to BLEU score. Experiments show that we can achieve higher performance using such an optimisation procedure and our word alignment approach is more flexible in a PB-SMT framework.

2 Syntactically Enhanced Word Alignment Model

2.1 General Model

Given a source sentence $C = c_1^J$ that consists of J Chinese words $\{c_1, \cdots, c_J\}$ and target sentence $E = e_1^I$ which consists of I English words $\{e_1, \cdots, e_I\}$, we seek to find the optimal alignment $\hat{A}$ such that:

$$\hat{A} = \underset{A}{\operatorname{argmax}} P(A|c_1^J, e_1^I)$$

We use a model (1) that directly models the linkage between source and target words similarly to (Ittycheriah and Roukos, 2005). The Chinese-to-English word alignment $A_{C \to E} = \{i|a_j = i\}$ is modelled as shown in (1). We decompose this model into an emission model and a transition model (4). The emission model can be further decomposed into an anchor alignment model (2) and a syntactically enhanced model (3) by distinguishing the anchor alignment from the non-anchor alignment.

$$\begin{aligned} p(A|c_1^J, e_1^I) &= \prod_{j=0}^{J} p(a_j|c_1^J, e_1^I, a_1^{j-1}) && (1) \\ &= \frac{1}{Z} \cdot p_\epsilon(A_\Delta|c_1^J, e_1^I) \cdot && (2) \\ &\quad \prod_{j \in \bar{\Delta}} p(a_j|c_1^J, e_1^I, a_1^{j-1}, A_\Delta) && (3) \\ &\quad \prod_{j=1}^{J} p(a_j|a_{j-1}, A_\Delta) && (4) \end{aligned}$$

2.2 Emission Model

2.2.1 Anchor Word Alignment

The anchor alignment model $p_\epsilon(A_\Delta)$ aims to find a set of high-precision links. Various approaches can be used for this purpose.

We can use the asymmetric IBM models for bidirectional word alignment and derive the intersection. Using this approach, we can obtain a set of anchor alignments $A_\Delta = \{i|i \in \Delta\}$. Subsequently, the anchor model is estimated as follows:

$$p(a_j) = \begin{cases} \alpha & \text{if } a_j = i \text{ and } i \in \Delta, \\ \frac{1-\alpha}{I} & \text{otherwise.} \end{cases}$$

The parameter α can be optimised on the development set. In our experiments we set $\alpha = 0.9$.

2.2.2 Syntactically Enhanced Word Alignment

The syntactically enhanced model is used to model the alignment of the words left unaligned after anchoring. We directly model the linkage between source and target words using a discriminative word alignment framework where various features can be incorporated. Given a source word c_j and the target sentence e_1^I, we search for the alignment a_j such that:

$$\begin{aligned} \hat{a_j} &= \underset{a_j}{\operatorname{argmax}} \{p_{\lambda_1^M}(a_j|c_1^J, e_1^I, a_1^{j-1}, A_\Delta)\} && (5) \\ &= \underset{a_j}{\operatorname{argmax}} \{\sum_{m=1}^{M} \lambda_m h_m(c_1^J, e_1^I, a_1^j, A_\Delta, T_c, T_e)\} \end{aligned}$$

In this decision rule, we assume that a set of highly reliable anchor alignments A_Δ has been obtained, and T_c (resp. T_e) is used to denote the dependency structure for source (resp. target) language. In such a framework, various machine learning techniques can be used for parameter estimation. The feature functions we used are described in section 3.

2.3 Transition Model

Given the anchor alignment, the first-order transition probability model (4) can be defined as follows:

$$p(a_j|a_{j-1}, A_\Delta) = \begin{cases} 1.0 & \text{if} \in \Delta, \\ \hat{p}(a_j|a_{j-1}) & \text{otherwise.} \end{cases}$$

Such a definition implies that an anchor alignment is always believed to be a correct alignment, maximum likelihood estimates obtained on a gold-standard word alignment corpus are used when the current word f_j is not involved in an anchor alignment. The estimation of $\hat{p}(a_j|a_{j-1})$ is calculated following the homogeneous HMM model (Vogel et al., 1996). Under this model, we assume that the

probability depends only on the jump width $(i-i')$, in order to make the alignment parameters independent of absolute word positions. Using a set of non-negative parameters $\{c(i-i')\}$, the transition probability can be written in the form:

$$p(a_j|a_{j-1}, A_\Delta) = \frac{c(i-i')}{\sum_{i''=1}^{I} c(i''-i')}$$

We use the refined model which extends the HMM network with I empty words e_{I+1}^{2I} and adds parameter p_0 to account for the transition probability to empty words (Och and Ney, 2003).

If a zero-order dependence is assumed in a transition model, the emission models is the only information to guide the word alignment.

2.4 Model Interpolation

We interpolate the general alignment model (1) as follows:

$$\begin{aligned} p(A|c_1^J, e_1^I) &= \frac{1}{Z} \cdot p_\epsilon(A_\Delta|c_1^J, e_1^I)^{1-\lambda} \cdot \\ &\prod_{j \in \bar{\Delta}} p(a_j|c_1^J, e_1^I, a_1^{j-1}, A_\Delta)^{1-\lambda} \cdot \\ &\prod_{j=1}^{J} p(a_j|a_{j-1}, A_\Delta)^{\lambda} \end{aligned}$$

We can use factor λ to weight the emission model and transition model probabilities so that the system can be optimised according to different objectives.

3 Feature Functions for Syntactically Enhanced Word Alignment

The various features used in our syntactically enhanced model can be classified into two groups: statistics-based features and syntactic features which are similar to those in (Ma et al., 2008)

3.1 Statistics-based Features

The statistics-based features we used include IBM model 1 score, Log-likelihood ratio (Dunning, 1993) and POS translation probability. We choose these features because they are empirically proven to be effective in word alignment tasks (Melamed, 2000; Liu et al., 2005; Moore, 2005).

3.2 Syntactic Features

The dependency relation R_e (resp. R_c) between two English (resp. Chinese) words e_i and $e_{i'}$ (resp. c_j and $c_{j'}$) in the dependency tree of the English sentence e_1^I (resp. Chinese sentence c_1^J) can be represented as a triple $<e_i, R_e, e_{i'}>$ (resp. $<c_j, R_c, e_{j'}>$). Given c_1^J, e_1^I and their syntactic dependency trees $T_{c_1^J}$, $T_{e_1^I}$, if e_i is aligned to c_j and $e_{i'}$ aligned to $c_{j'}$, according to the dependency correspondence assumption (Hwa et al., 2002), there exists a triple $<c_j, R_c, c_{j'}>$.

While we are not aiming to justify the feasibility of the dependency correspondence assumption by proving to what extent $R_e = R_c$ under the condition described above, we do believe that c_j and $c_{j'}$ are likely to be dependent on each other. Given the anchor alignment A_Δ, a candidate link (j, i) and the dependency trees, we can design four classes of feature functions.

3.2.1 Agreement features

The agreement features can be further classified into dependency agreement features and dependency label agreement features. Given a candidate link (j, i) and the anchor alignment A_Δ, the dependency agreement (DA) feature function is defined as follows:

$$h_{DA-1} = \begin{cases} 1 & \text{if } \exists <c_j, R_c, c_{j'}>, <e_i, R_e, e_{i'}> \\ & \text{and } (j', i') \in A_\Delta, \\ 0 & \text{otherwise.} \end{cases}$$

By changing the dependency direction between the words c_j and $c_{j'}$, we can derive another dependency agreement feature:

$$h_{DA-2} = \begin{cases} 1 & \text{if } \exists <c_{j'}, R_c, c_j>, <e_{i'}, R_e, e_i> \\ & \text{and } (j', i') \in A_\Delta, \\ 0 & \text{otherwise.} \end{cases}$$

We can define the dependency label agreement feature[1] as follows:

$$h_{DLA-1} = \begin{cases} 1 & \text{if } \exists <c_j, R_c, c_{j'}>, <e_i, R_e, e_{i'}> \\ & \text{and } (j', i') \in A_\Delta, R_c = R_e, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly we can obtain h_{DLA-2} by changing the dependency direction.

3.2.2 Source word dependency features

Given a candidate link (j, i) and anchor alignment A_Δ, source language dependency features are used to capture the dependency label between a

[1]Note that we used the same dependency parser for source and target language parsing.

source word c_j and a source anchor word $c_k \in \Delta$. For example, a feature function relating to dependency type 'PRD' can be defined as:

$$h_{src-1-PRD} = \begin{cases} 1 & \text{if } \exists < c_j, R_c, c_{j'} > \\ & \text{and } R_c = \text{'PRD'}, \\ 0 & \text{otherwise.} \end{cases}$$

By changing the direction we can obtain $h_{src-2-PRD}$.

3.2.3 Target word dependency features

Target word dependency features can be defined in a similar way as source word dependency features.

3.2.4 Target anchor feature

The target anchor feature defines whether the target word e_i is an anchor word.

$$h_{src-1-PRD} = \begin{cases} 1 & \text{if } i \in a_\Delta, \\ 0 & \text{otherwise.} \end{cases}$$

4 Experimental Setting

4.1 Data

The experiments were carried out using the Chinese–English datasets provided within the IWSLT 2007 evaluation campaign (Fordyce, 2007). We tagged all the sentences in the training and devset3 using a maximum entropy-based POS tagger, namely MXPOST (Ratnaparkhi, 1996), trained on the Penn English and Chinese Treebanks. Both Chinese and English sentences are parsed using the Malt dependency parser (Nivre et al., 2007), which achieved 84% and 88% labelled attachment scores for Chinese and English (each has 11 dependency labels) respectively.

4.1.1 Word Alignment

We manually annotated word alignments on devset3. Following recent research in measuring word alignment quality for SMT purposes, we set all the word alignment links as sure (S) links (cf. (Fraser and Marcu, 2007)). IWSLT devset3 consists of 502 sentence pairs after cleaning. We used the first 300 sentence pairs for training, the following 50 sentence pairs as validation set and the last 152 sentence pairs for testing. The various statistics for the gold-standard corpus is listed in Table 1.

		Chinese	English
Train	Sentences	300	
	Running words	2,231	2,704
	Vocabulary size	636	709
	Sure links	2773	
Dev.	Sentences	50	
	Running words	445	451
	Vocabulary size	205	212
	Sure links	555	
Eval.	Sentences	152	
	Running words	1,107	1,149
	Vocabulary size	394	413
	Sure links	1400	

Table 1: Chinese–English word alignment gold-standard corpus statistics

4.1.2 Machine Translation

Training was performed using the default training set (39,952 sentence pairs), to which we added the set devset1 (506 sentence pairs) and devset2 (500 sentence pairs).[2] We used devset4 (489 sentence pairs, 7 references) to tune various parameters in the MT system and IWSLT 2007 test set (489 sentence pairs, 6 references) for testing. Detailed corpus statistics are shown in Table 2.

		Chinese	**English**
Train	Sentences	40,958	
	Running words	357,968	385,065
	Vocabulary size	11,362	9,718
Dev.	Sentences	489 (7 ref.)	
	Running words	5,717	46,904
	Vocabulary size	1,143	1,786
Eval.	Sentences	489 (7 ref.)/489 (6 ref.)	
	Running words	3,166	23,181
	Vocabulary size	862	1,339

Table 2: Corpus statistics IWSLT 2007 data set

4.2 Alignment Training and Decoding

In our experiments, we treated anchor alignment and syntactically enhanced alignment as separate processes in a pipeline. The anchor alignments are kept fixed so that the parameters in the syntactically enhanced model can be optimised.[3] We used the support vector machine (SVM) toolkit, SVM_light[4] to optimise the parameters in (5). Our model is constrained in such a way that each

[2]More specifically, we chose the first English reference from the 16 references and the Chinese sentence to construct new sentence pairs.

[3]Note that our anchor alignment does not achieve 100% precision. Since we performed precision-oriented alignment for the anchor alignment model, the errors in anchor alignment will not bring much noise into the syntactically enhanced model.

[4]http://svmlight.joachims.org/

source word can only be aligned to one target word.

In SVM training, we transform each possible link involving the words left unaligned after anchoring into an event. Positive examples (aligned pairs) are assigned the target value 1 and negative examples (unaligned pairs) assigned -1. Using this training data, we can build a regression model to estimate the reliability of alignment given a pair of words. The value of functional margin obtained by applying the regression model serves as the emission probability in our word alignment model.

For the first-order transition model, we estimate the transition probability on a gold-standard word alignment corpus in training. In decoding, the best alignment path is searched out using a Viterbi-style decoding algorithm. The interpolation factor λ can be optimised on development set. When a zero-order transition model (a uniform transition distribution) is used, we constrain the emission probability by a threshold t, which is set as the minimal reliability score for each link. Again, t can be optimised according to the development set.

The decoding is performed separately in two directions (Chinese-to-English and English-to-Chinese), and we then obtain the refined alignments as the final word alignment.

4.3 Baselines

4.3.1 Word Alignment

We used the GIZA++ implementation of IBM word alignment model 4 (Brown et al., 1993; Och and Ney, 2003) for word alignment, and the heuristics described in (Koehn et al., 2003) to derive the intersection and refined alignment.

4.3.2 Machine Translation

We use a standard log-linear PB-SMT model as a baseline: GIZA++ implementation of IBM word alignment model 4,[5] the refinement and phrase-extraction heuristics described in (Koehn et al., 2003), minimum-error-rate training (Och, 2003), a trigram language model with Kneser-Ney smoothing trained with SRILM (Stolcke, 2002) on the English side of the training data, and Moses (Koehn et al., 2007) to decode.

[5]More specifically, we performed 5 iterations of Model 1, 5 iterations of HMM, 3 iterations of Model 3, and 3 iterations of Model 4.

4.4 Evaluation

We evaluate the intrinsic quality of the predicted alignment A with Precision, Recall and the balanced F-score with $\alpha = 0.5$ (cf. (Fraser and Marcu, 2007)).

$$\text{Recall} = \frac{|A \cap S|}{|S|} \quad \text{Precision} = \frac{|A \cap S|}{|A|}$$

$$\text{F-score}(A, S, \alpha) = \frac{1}{\frac{\alpha}{Precision(A,S)} + \frac{1-\alpha}{Recall(A,S)}}$$

Research has shown that an increase in AER does not necessarily imply an improvement in translation quality (Liang et al., 2006) and vice-versa (Vilar et al., 2006). Hereafter, we use a Chinese–English MT task to extrinsically evaluate the quality of our word alignment. The translation output is measured using BLEU (Papineni et al., 2002).

5 Experiments

5.1 Word Alignment Results

We performed word alignment bidirectionally using our approach to obtain the refined alignments (Koehn et al., 2003) and compared our results with two strong baselines based on generative word alignment models. The results are shown in Table 3. We can see that both the syntactically enhanced model based on HMM intersection anchors and on IBM model 4 anchors achieved higher F-scores than the pure generative word alignment models. It is also can be seen that zero-order syntactic models are better in precision and first-order models are superior in recall. The best result achieved 2.99% relative increase in F-score compared to the baseline when we use IBM model 4 intersection to obtain the set of anchor alignments.

model	Precision	Recall	F-score
Model 1	65.98	70.64	68.23
+Syntax-zero-order	**80.71**	69.93	**74.93**
+Syntax-first-order	72.84	**73.36**	73.10
HMM refined	73.80	73.86	73.83
+Syntax-zero-order	**83.65**	70.14	76.30
+Syntax-first-order	77.17	**76.07**	**76.62**
Model 4 refined	75.87	78.14	76.99
+Syntax-zero-order	**84.59**	74.50	**79.29**
+Syntax-first-order	80.21	**77.57**	78.87

Table 3: The performance of our syntactically enhanced word alignment approach

5.2 Machine Translation Results

Table 4 shows the influence of our word alignment approach on MT quality.[6] From Table 4, we can see that our zero-order syntactically enhanced model based on Model 4 anchors achieved 1.82 absolute BLEU score (5.38% relative) improvement compared to its baseline counterpart on the test set, which is statistically significant ($p<0.002$) using approximate randomisation (Noreen, 1989) for significance testing. However, the first-order model suffers from overfitting problems, with a significant improvement on the development set and no improvement on the test set.

	dev	test
Baseline-Model4	24.13	33.85
+Syntax-zero-order	25.41	**35.67**
+Syntax-first-order	25.47	33.70

Table 4: Syntactically enhanced word alignment for PB-SMT optimised according to BLEU

5.2.1 Different Optimisation Criteria

The parameter t (threshold) for zero-order models can be optimised with either F-score (OFscore) obtained on a gold-standard word alignment corpus, or BLEU score (OBLEU) on a development set of an MT system as the objective. Similarly for first-order models, parameters λ and p_0 can be optimised according to these two criteria. Given that we have a very limited number of parameters to optimise (just two, i.e. $t_{c\to e}$ for Chinese–English and $t_{e\to c}$ for English–Chinese in the zero-order model, and three parameters, i.e. $\lambda_{c\to e}$, $\lambda_{e\to c}$ and p_0 in the first-order model), we used a simple greedy search algorithm by search a predefined set of possible parameter settings. For example, we tried different value combinations from the set $\{-1.7, -1.6, \cdots, 0.0\}$ for $t_{c\to e}$ and for $t_{c\to e}$. Table 5 shows the results according to different optimisation criteria using Model 4 intersected alignments as anchors.

For the zero-order model, the best parameter set is $t_{c\to e} = -1.0$ and $t_{e\to c} = -0.6$ according to F-score; however, according to BLEU, the best parameters are $t_{c\to e} = -0.8$ and $t_{e\to c} = -0.9$. From Table 5, we can see that the BLEU score obtained when word alignment is optimised according to F-score is slightly inferior (not statistically significant) to that when optimised according to BLEU. The search graph of optimisation according to BLEU is shown in Figure 2. The different optimisation criteria do not have much impact on the F-score. For the first-order model, the best parameter setting is $\lambda_{c\to e} = 0.2$, $\lambda_{e\to c} = 0.2$ and $p_0 = 0.6$ according to F-score. However, according to BLEU, it is $\lambda_{c\to e} = 0.9$, $\lambda_{e\to c} = 0.3$ and $p_0 = 0.8$. From Table 5, we can observe that parameters optimised according to BLEU suffer from overfitting. The word alignment optimised according to F-score not only yields a higher F-score, but also achieves better performance on the test set when used in a PB-SMT system.

[6]Note that the only difference between our MT system and the baseline PB-SMT system is the word alignment component.

		BLEU		F-score	
		dev	test	dev	test
Zero-order	OFscore	24.74	35.21	*77.49*	79.23
	OBLEU	25.41	**35.67**	76.98	*79.25*
First-order	OFscore	23.75	34.32	76.41	78.87
	OBLEU	**25.47**	33.70	70.75	72.33

Table 5: Optimising syntactically enhanced word alignment for PB-SMT

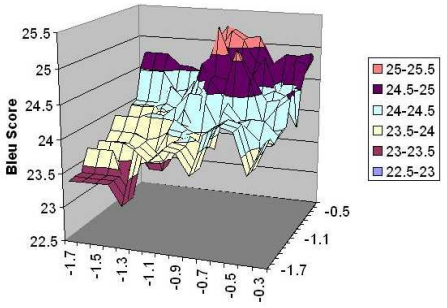


Figure 2: Search graph obtained when optimising BLEU

5.2.2 Phrase Extraction

To further investigate the impact of our word alignment on SMT, we compared the extracted phrase table using our word alignment against the baseline phrase table. Figure 3 shows the size of the phrase tables when the system use different word alignment. We observed that using the zero-order syntactically enhanced word alignment tends to extract fewer phrase pairs (more word alignment links) when optimised according to BLEU. As an exception, the first-order word alignment which suffered from overfitting extracted far more phrase pairs (fewer word alignment links) when optimised according to BLEU. All syntactically enhanced word alignments lead to larger phrase tables.

Figure 3: A comparison of the number of phrase pairs

5.2.3 Scaling Up

To test the scalability of our approach, we added in a further 130K sentence pairs from HIT corpus provided for IWSLT 2008 evaluation campaign. We re-use the parameters obtained from the IWSLT 2007 corpus in these experiments. Table 6 shows the results. For the zero-order syntactically enhanced model optimised according to BLEU, we observed an increase of 1.69 absolute BLEU scores over the baseline on the development set; on the test set, however, no improvement was achieved. For the first-order model, given the parameters we obtained on IWSLT 2007 data set by optimising BLEU suffered from overfitting, the consequence can also be seen on the experiments using the larger data set. From these results, we can see the limitation of the optimisation process and a more informative objective function is needed to achieve better performance.

		dev	test
Baseline-Model4		27.05	**35.65**
Syntax-zero-order	OFscore	26.93	35.35
	OBLEU	**28.74**	35.47
Syntax-first-order	OFscore	27.05	35.16
	OBLEU	28.17	34.95

Table 6: Scaling up syntactically enhanced word alignment for PB-SMT

6 Comparison with Previous Work

(Fraser and Marcu, 2007) proposed a semi-supervised model that can take advantage of both generative and discriminative models. However, in their model word alignment is still a standalone component in PB-SMT and cannot be tuned for PB-SMT performance. (Lambert et al., 2007) attempted to tune a discriminative word alignment model directly with MT in mind. Our work investigates the tuning of word alignment that takes advantage of both generative and discriminative word alignment models. (Ma et al., 2008) proposed a similar word alignment framework; however, their word alignment was only tuned according to AER and the improvement for PB-SMT system was not statistically significant. We show that by tuning word alignment according to PB-SMT performance, we can achieve significantly better results.

7 Conclusions and Future Work

In this paper, we proposed a flexible syntactically enhanced word alignment model that can be tuned according to different end tasks. This model takes the advantages of both unsupervised and supervised word alignment approaches by obtaining anchor alignments from unsupervised generative models and seeding the anchor alignments into a supervised discriminative model. This model offers the flexibility of tuning the alignment according to different optimisation criteria.

Our model is superior to generative word alignment models in terms of both intrinsic and extrinsic quality. We observed a 2.99% relative increase in F-score compared to the best baseline system. Using our word alignment in a PB-SMT system yields a 5.38% relative increase in BLEU score.

In the future, we first plan to conduct an in-depth investigation regarding what type of word alignments are beneficial to MT. We also plan to refine the optimisation criteria to avoid overfitting problems. Finally, we will conduct experiments in other domains and on other language pairs.

Acknowledgement

This work is supported by Science Foundation Ireland (O5/IN/1732 and 07/CE/I1142) and the Irish Centre for High-End Computing.[7] We would like to thank the reviewers for their insightful comments.

References

Brown, Peter F., 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.

Deng, Yonggang and William Byrne. 2006. MTTK: An alignment toolkit for statistical machine transla-

[7] http://www.ichec.ie/

tion. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 265–268, New York City, NY.

Dunning, Ted. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Fordyce, Cameron Shaw. 2007. Overview of the IWSLT 2007 Evaluation Campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 1–12, Trento, Italy.

Fraser, Alexander and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303.

Hwa, Rebecca, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 392–399, Philadelphia, PA.

Ittycheriah, Abraham and Salim Roukos. 2005. A maximum entropy word aligner for Arabic-English machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 89–96, Vancouver, British Columbia, Canada.

Koehn, Philipp, Franz Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 48–54, Edmonton, Canada.

Koehn, Philipp, 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 *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Lambert, Patrik, Rafael E. Banchs, and Josep M. Crego. 2007. Discriminative alignment training without annotated data for machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 85–88, Rochester, NY.

Liang, Percy, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 104–111, New York, NY.

Liu, Yang, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 459–466, Ann Arbor, MI.

Ma, Yanjun, Sylwia Ozdowska, Yanli Sun, and Andy Way. 2008. Improving word alignment using syntactic dependencies. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 69–77, Columbus, OH.

Melamed, I. Dan. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

Moore, Robert C. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Vancouver, BC, Canada.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Ervin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Noreen, Eric W. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience, New York, NY.

Och, Franz and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, Franz. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Papineni, Kishore, 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*, pages 311–318, Philadelphia, PA.

Ratnaparkhi, Adwait. 1996. A maximum entropy model for part-of-speech tagging. In Brill, Eric and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, Somerset, NJ.

Stolcke, Andrea. 2002. SRILM – An extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, Denver, CO.

Vilar, David, Maja Popovic, and Hermann Ney. 2006. AER: Do we need to "improve" our alignments? In *Proceedings of the International Workshop on Spoken Language Translation*, pages 205–212, Kyoto, Japan.

Vogel, Stefan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark.

Author Index