Ph.D. thesis

Using unsupervised corpus-based methods to build rule-based machine translation systems

Tesis doctoral

Empleo de métodos no supervisados basados en corpus para construir traductores automáticos basados en reglas

Felipe Sánchez Martínez

Supervised by

Dirigida por

Mikel L. Forcada

Juan Antonio Pérez Ortiz

Universitat d'Alacant
Universidad de Alicante

Departament de Llenguatges i Sistemes Informàtics
Departamento de Lenguajes y Sistemas Informáticos

May 2008

A la memoria de mi madre,
Mª del Carmen.

Agradecimientos

En primer lugar, quiero agradecer a mis dos directores de tesis, Mikel L. Forcada y Juan Antonio Pérez Ortiz, su apoyo e inestimable ayuda a lo largo de la elaboración de esta tesis; sin su constante supervisión y crítica constructiva esta tesis nunca hubiera llegado a buen término. También quiero agradecer a Rafael C. Carrasco sus comentarios acerca del método de entrenamiento de desambiguadores léxicos categoriales para traducción automática, y que me ofreciera la oportunidad de compaginar la finalización de esta tesis con el trabajo como técnico en el proyecto que dirige.

Quiero agradecer también a mis compañeros de trabajo del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante su apoyo y ánimo. Especialmente a Sergio, por escucharme y darme su opinión sobre lo que tuviera entre manos todas las veces que he entrado en su despacho a interrumpirle casi sin llamar; a Gema, por más de lo mismo y por aguantar con una sonrisa todos los *tesinazos* que le he dado; a Marcel, con quien comparto despacho, por estar ahí sentado y darme ánimo y un ambiente de trabajo envidiable. Tampoco puede faltar mi agradecimiento a *os brasileiros*: Helena, Thiago, Graça, Anselmo, Ariani, Thiago Carbonell y Jorge, por su cálida acogida durante mi estancia en Brasil.

Gracias a la familia por su apoyo, a mi padre y hermanos por escucharme y hacer como que entienden de qué va esta tesis y especialmente a mis sobrinos, Ángela y Pablo, que, aunque ellos no lo sepan, con su sonrisa me han hecho más llevadero todo el trabajo y esfuerzo que ha supuesto esta tesis. También a los *amigotes* (nos los nombro por si se me olvida alguno y me canean) por esos momentos de *desconexión*, que no todo puede ser trabajo. Y, aunque lo deje para el final, no por ello es menos importante, gracias a Elisa, por animarme, aguantar mis malos momentos, los buenos también, por todo.

Por último gracias al Ministerio de Educación y Ciencia y al Fondo Social Europeo por la beca de Formación de Personal Investigador (FPI) con referencia BES-2004-4711 y al Ministerio de Industria, Turismo y Comercio por el proyecto TIC2003-08681-C02-01 al que se encontraba adscrita dicha beca. Gracias también a los distintos proyectos en los que he trabajado y que me han brindado la oportunidad de mejorar y experimentar con más pares de lenguas: los proyectos FIT340101-2004-3 y FIT-350401-2006-5 financiados por el Ministerio de Industria, Turismo y Comercio en el marco de los cuales he participado en el desarrollo de la plataforma de traducción automática de código abierto Apertium; el proyecto *Traducció automàtica de codi obert per al català* de la

Generalitat de Catalunya para el desarrollo de los pares de lenguas occitano–catalán y francés–catalán; y por último, el proyecto del Ministerio de Educación y Ciencia TIN2006-15071-C03-01 en el que actualmente trabajo.

Felipe Sánchez Martínez
Alicante, 5 de mayo de 2008

Preface

During the last years, corpus-based approaches to machine translation (MT), such as statistical MT or example-based MT have grown in interest as a consequence of the increasing availability of bilingual texts in electronic format. However, corpus-based approaches are not applicable when the translation involves less-resourced language pairs for which there are no *parallel* corpora available, or the size of such corpora is not large enough to build a general-purpose MT system; in those cases, the rule-based approach is the only applicable solution. This is currently the case of less-resourced language pairs such as Occitan–Catalan, French–Catalan or English–Afrikaans, among others.

Since I started to work in 2003 at the Departament de Llenguatges i Sistemes Informàtics at Universitat d'Alacant[4] I have participated in the development of rule-based MT (RBMT) systems such as the Spanish–Catalan MT system interNOSTRUM,[5] the Spanish–Portuguese MT system Traductor Universia[6] and the open-source shallow-transfer MT platform Apertium,[7] which has several language pairs available. In the development of all these RBMT systems I mainly focused on the development of the part-of-speech taggers, although I also participated in the overall design of the Apertium architecture. Experience in the development of MT systems of this kind has shown to me the huge human effort that involves coding all the linguistic resources needed to build them.

This thesis focuses on the development of unsupervised methods to obtain automatically from corpora some of the linguistic resources required to build RBMT systems; more precisely, shallow-transfer MT systems like those in whose development I have been involved. Specifically, this thesis focuses on: (i) an unsupervised method to train part-of-speech (PoS) taggers to be used in RBMT; (ii) the automatic inference of the set of states to be used by PoS taggers based on hidden Markov models for use in RBMT; and, (iii) the automatic inference of shallow-transfer rules from a small amount of parallel corpora. The final goal is to reduce as much as possible the human effort needed to build an RBMT system from scratch.

[4] `http://www.dlsi.ua.es`
[5] `http://www.internostrum.com`
[6] `http://traductor.universia.net`
[7] `http://apertium.sf.net`

The approaches that will be discussed in this thesis will show that to train PoS taggers based on hidden Markov models (HMM) —in an unsupervised way— there is a source of knowledge, namely, a statistical model of the target language, that can be easily used to produce PoS taggers specially suited for use in RBMT. In addition, it will show how to apply a clustering algorithm to automatically determine the set of hidden states to be used by HMM-based PoS taggers. Finally, this thesis will demonstrate that shallow structural transfer rules can be inferred from a small amount of parallel corpora by using alignment templates like those used in statistical MT.

All the approaches and methods that will be discussed in this thesis have been implemented and released as open source in order to allow the whole community to benefit from them; moreover, they have been implemented as tools for the development of new language pairs for Apertium. The public availability of the source code guarantees the reproducibility of all the experiments conducted. It also allows other researchers to improve them and saves the time and effort of people developing new language pairs for Apertium.

This thesis has been possible thanks to the ideas and constant supervision of Drs. Mikel L. Forcada and Juan Antonio Pérez-Ortiz from the Departament de Llenguatges i Sistemes Informàtics at Universitat d'Alacant. Nevertheless, the part of this thesis that deals with the inference of shallow-transfer rules by adapting the alignment template approach was initially developed during my three-months stay in 2005 at the Chair of Computer Science 6 (Computer Science Department) at the RWTH Aachen University[8] (Germany) under the supervision of Dr. Hermann Ney. The approach was later improved thanks to suggestions by Dr. Mikel L. Forcada.

Structure of this thesis

This thesis is structured in 6 chapters and 3 appendices. Here is a brief abstract of each one:

Chapter 1 begins by giving an introduction to MT and to the different approaches that can be followed to tackle the MT problem. Then, it explains the problems addressed in this thesis, the solutions to them that can be found in the literature and a brief outline of the approaches that will be discussed in the following chapters.

Chapter 2 presents a new unsupervised method that can be used to train HMM-based PoS taggers to be used in RBMT. The method uses information from the target language and from the remaining modules of the MT platform in which the PoS tagger will be integrated. The experiments show that this method produces better

[8] `http://www-i6.informatik.rwth-aachen.de`

PoS taggers to be used in RBMT than the standard unsupervised algorithm to train HMM-based PoS taggers.

Chapter 3 introduces an approach that can be used to speed up the PoS tagger training method described in Chapter 2 without affecting the quality achieved by the resulting PoS tagger.

Chapter 4 describes a clustering algorithm to be applied over the states of an initial HMM-based PoS tagger to reduce the final number of states of the HMM and, consequently, the number of parameters to estimate.

Chapter 5 explains a method to infer shallow-transfer rules from a small amount of parallel data by extending the alignment template approach used in statistical MT. The experiments will show that the inferred rules produce translations whose quality is close to that achieved by hand-coded transfer rules.

Chapter 6 summarizes the main contributions of this thesis and outlines some future research lines.

Appendix A explains in detail the Apertium open-source shallow-transfer MT platform. This MT platform, in whose development I have participated, is used along the whole thesis to test the approaches presented in the different chapters.

Appendix B overviews the use of HMMs for PoS tagging and summarizes classical supervised and unsupervised algorithms to train HMMs.

Appendix C describes the open-source software released as part of this thesis.

Publications

Some parts of this thesis have been published in journals, conference and workshop proceedings. Here is a list of papers in chronological order (in brackets, the chapter, or chapters, to which each paper is related):

- Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, Mikel L. Forcada (2004b). Exploring the use of target-language information to train the part-of-speech tagger of machine translation systems. In *Lecture Notes in Computer Science 3230 (Advances in Natural Language Processing, Proceedings of EsTAL - España for Natural Language Processing)*, p. 137–148, October 20–22, Alacant, Spain (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez04b.pdf`). [**Chapter 2**]

- Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, Mikel L. Forcada (2005). Target-language-driven agglomerative part-of-speech tag clustering for machine translation. In *Proceedings of the International Conference on Recent Advances*

in Natural Language Processing (RANLP 2005), p. 471–477, September 21–23, Borovets, Bulgaria (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez05.pdf`). [**Chapter 4**]

- Felipe Sánchez-Martínez, Hermann Ney (2006). Using alignment templates to infer shallow-transfer machine translation rules. In *Lecture Notes in Computer Science 4139 (Advances in Natural Language Processing, Proceedings of FinTAL 2006, 5th International Conference on Natural Language Processing)*, p. 756–767, August 23–25, Turku, Finland (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez06a.pdf`). [**Chapter 5**]

- Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, Mikel L. Forcada (2006). Speeding up target-language driven part-of-speech tagger training for machine translation. In *Lecture Notes in Computer Science 4293 (Advances in Artificial Intelligence, Proceedings of MICAI 2006, 5th Mexican International Conference on Artificial Intelligence)*, p. 844–854, November 13-17, Apizaco, Mexico (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez06b.pdf`). [**Chapter 3**]

- Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, Mikel L. Forcada (2007b). Integrating corpus-based and rule-based approaches in an open-source machine translation system. In *Proceedings of METIS-II Workshop: New Approaches to Machine Translation, a workshop at CLIN 17 - Computational Linguistics in the Netherlands*, p. 73–82, January 11, Leuven, Belgium (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez07a.pdf`). [**Chapters 1, 2 and 5**]

- Felipe Sánchez-Martínez, Carme Armentano-Oller, Juan Antonio Pérez-Ortiz, Mikel L. Forcada (2007a). Training part-of-speech taggers to build machine translation systems for less-resourced language pairs. In *Procesamiento del Lenguaje Natural nº 39, (XXIII Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural)*, p. 257–264, September 10–12, Sevilla, Spain (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez07b.pdf`). [**Chapter 2**]

- Felipe Sánchez-Martínez, Mikel L. Forcada (2007). Automatic induction of shallow-transfer rules for open-source machine translation. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007)*, p. 181–190, September 7–9, Skövde, Sweden (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez07c.pdf`). [**Chapter 5**]

- Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, Mikel L. Forcada (2008). Using target-language information to train part-of-speech taggers for machine translation. In *Machine Translation*, 22(1-2):29–66. (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez08b.pdf`). [**Chapters 2 and 3**]

- Felipe Sánchez-Martínez, Mikel L. Forcada (2009). Inferring shallow-transfer machine translation rules from small parallel corpora. In *Journal of Artificial Intelligence Research*, 34:605–635 (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez-martinez09b.pdf`). [**Chapter 5**]

There are other papers I have published in conference proceedings which, even if not directly related to the work of this thesis, relate to the Apertium open-source shallow-transfer MT engine, or to the other shallow-transfer MT systems previously mentioned (interNOSTRUM and Traductor Universia):

- Patrícia Gilabert-Zarco, Javier Herrero-Vicente, Sergio Ortiz-Rojas, Antonio Pertusa-Ibáñez, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Marcial Samper-Asensio, Míriam A. Scalco, Mikel L. Forcada (2003). Construcción rápida de un sistema de traducción automática español-portugués partiendo de un sistema español-catalán. In *rocesamiento del Lenguaje Natural, (XIX Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural)*, p. 279–284, September 10–12, Alcalá de Henares, Spain (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/gilabert03p.pdf`).

- Alicia Garrido-Alenda, Patrícia Gilabert-Zarco, Juan Antonio Pérez-Ortiz, Antonio Pertusa-Ibáñez, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Míriam A. Scalco, Mikel L. Forcada (2004). Shallow parsing for Portuguese-Spanish machine translation. In *Language technologies for Portuguese: shallow processing tools and resources*, p. 135–144, ed. Colibri, Lisbon, Portugal (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/garrido04p.pdf`).

- Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, Mikel L. Forcada (2004a). Cooperative unsupervised training of the part-of-speech taggers in a bidirectional machine translation system. In *Proceedings of The 10th Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2004)*, p. 135–144, October 4–6, Baltimore, MD USA (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/sanchez04a.pdf`).

- Antonio M. Corbí-Bellot, Mikel L. Forcada, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Iñaki Alegria, Aingeru Mayor, Kepa Sarasola (2005). An open-source shallow-transfer machine translation engine for the Romance languages of Spain. In *Proceedings of the Tenth Conference of the European Association for Machine Translation*, p. 79–86, May 30–31, Budapest, Hungary (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/corbi05.pdf`).

- Carme Armentano-Oller, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Boyan Bonev, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez (2005). An open-source shallow-transfer machine translation toolbox: consequences of its release and availability. In *OSMaTran: Open-Source Machine Translation, A workshop at Machine Translation Summit X*, p. 23–30, September 12–16, Phuket, Thailand (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/armentano05p.pdf`).

- Carme Armentano-Oller, Rafael C. Carrasco, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz,

Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Miriam A. Scalco (2006). Open-source Portuguese-Spanish machine translation. In *Lecture Notes in Computer Science 3960 (Computational Processing of the Portuguese Language, Proceedings of the 7th International Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR 2006)*, p. 50–59, May 13–17, Itatiaia, Rio de Janeiro, Brazil (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/armentano06.pdf`).

- Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Mikel L. Forcada (2006). Opentrad Apertium open-source machine translation system: an opportunity for business and research. In *Proceedings of Translating and the Computer 28 Conference*, November 16–17, London, United Kingdom (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/ramirez06.pdf`).

- Carme Armentano-Oller, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Marco A. Montava, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez (2007). Apertium, una plataforma de código abierto para el desarrollo de sistemas de traducción automática. In *Proceedings of FLOSS (Free/Libre/Open Source Systems) International Conference*, p. 5–20, March 7–9, Jerez de la Frontera, Spain (`http://www.dlsi.ua.es/~fsanchez/pub/pdf/armentano07.pdf`).

Contents

Chapter 1

Introduction

This thesis deals with common problems in natural language processing, and more precisely, in machine translation (MT). This chapter introduces basic concepts about MT and the different approaches that are usually followed to tackle the MT problem. Then it explains the problems addressed in this thesis, how they have been approached in the literature, and a brief outline of the approaches that will be discussed in this dissertation.

1.1 Machine translation

Machine translation (MT) may be defined as the use of a computer to translate a text from one natural language, the *source language* (SL), into another one, the *target language* (TL); although MT may involve the translation of speech, this work focuses only on the translation of texts that are supposed to be grammatical and consisting of well-formed sentences.

MT is difficult mainly because natural languages are highly ambiguous and because two languages are not always allowed to express the same content. Somers (2003) classifies the nature of the problems that make the MT task to be a difficult one into four groups:

1. *Form under-determines content.* It is not always possible to determine the content from what is written; in the sentence *"I saw the girl with the telescope"*, *"with the telescope"* can be either related to the action of seeing (*"saw"*), or to the grammatical object being observed (*"the girl"*).

2. *Content under-determines form.* It is difficult to know how some particular content should be expressed because there are many ways in which the same thing can be expressed in the same language.

3. *Languages differ.* Languages may use different structures to express the same meaning. Consider the English sentence *"I like apples"* and its translation into Spanish *"Me gustan las manzanas"*; while *apples* is a direct object, *manzanas* (the translation of *apples*) is the subject of the Spanish sentence.

4. *Translation is difficult to describe.* Translation involves a huge amount of human knowledge which must be manually described and coded in a usable form, or automatically learned from translation examples.

Concerning the applications of MT, there are two possible applications of a text-to-text MT system, namely assimilation and dissemination. In *assimilation* the goal is to obtain a translation in the TL that is understandable. A daily use of MT systems for assimilation happens when people surf the web in a foreign language; their objective may be to know if the web page, that has been automatically translated, contains the information they are looking for; no matter whether the translation is ungrammatical or it contains untranslated words if the reader is able to read it and to understand what the document is about.

When using MT systems for *dissemination* the aim is to produce a translation that is then *post-edited* (corrected) by human translators. This is the case, for example, of public institutions such as the European Union, which introduce laws in several languages, or newspapers published in more than one language, such as *El Periódico de Catalunya*.[1]

For easy post-edition, the MT output needs to contain translation errors that may be easily identified by human translators accustomed to the MT system being used; this implies a repetitive nature in the translation errors produced and an intelligible translation that may be post-edited without referring to the SL text.

1.2 Approaches to machine translation

The different ways in which the MT problem has been approached may be classified according to the nature of the knowledge used in the development of the MT system. From this viewpoint one can distinguish between corpus-based and rule-based approaches; although hybrid approaches between them are possible.

Corpus-based approaches to MT use large collections of parallel texts as the source of knowledge from which the engine learns how to perform translations. A *parallel text* is a text in one language together with its translation in another language; a large collection of parallel texts is usually referred to as a *parallel corpus*. To learn translations from a parallel corpus, the latter usually needs to be aligned at the sentence

[1] `http://www.elperiodico.com`

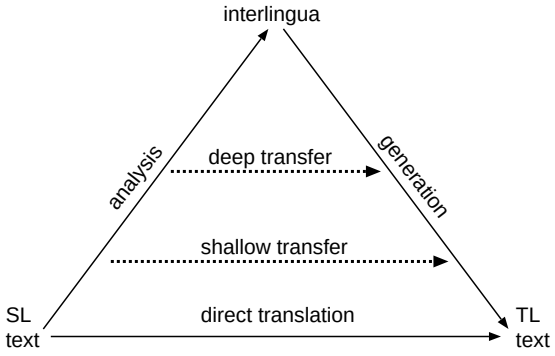


Figure 1.1: Vauquois Pyramid: Different levels of abstraction in RBMT.

level, that is, for each sentence in one language the counterpart sentence in the other language needs to be clearly identified. Obtaining the alignments at the sentence level is not trivial, as sentences may be deleted, inserted, split or merged during translation.

There are two main types of corpus-based approaches to the MT problem, *example-based* MT (EBMT) and *statistical* MT (SMT). In EBMT (Nagao, 1984; Carl and Way, 2003) the translation is performed by analogy: given an SL sentence and a parallel corpus, EBMT tries to find the "best" match for the SL sentence in the parallel corpus and retrieves its TL part as the translation. As matching whole sentences may not succeed, the SL sentence is analyzed into parts whose translations are retrieved from an analogously analyzed parallel corpus, and recombined to build the translated sentence in the TL. In SMT (Brown et al., 1993; Knight, 1999; Lopez, 2008) translations are generated on the basis of statistical translation models whose parameters are learned from parallel corpora.

Rule-based MT (RBMT) systems use knowledge in the form of rules explicitly coded by human experts that try to describe the translation process. This kind of MT systems heavily depend on linguistic knowledge such as morphological and bilingual dictionaries (containing lexical, syntactic and even semantic information), part-of-speech (PoS) disambiguation rules or manually disambiguated corpora, and a large set of rules; the process of building an RBMT system involves a huge human effort to build the necessary linguistic resources (Somers, 2003).

Generally, RBMT systems work by parsing (or analyzing) the SL text, usually creating a intermediate (symbolic) representation (IR), from which the text in the TL is generated (Hutchins and Somers, 1992). According to the nature of the IR used, an RBMT system may be said to be either an interlingua or a transfer-based MT system (see Vauquois Pyramid on Figure 1.1).

An *interlingua* MT system uses a single, language-independent IR. The advantage of using a language-independent IR is that no bilingual information (dictionaries or rules)

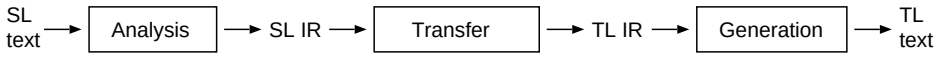


Figure 1.2: Scheme of a general transfer-based MT system.

are needed; as a disadvantage we have that the definition of a language-independent IR is very difficult, perhaps impossible for open-domain translations.

In *transfer-based* MT the IR depends on the languages involved in the translation. These systems usually work by applying a set of structural *transfer* rules to the SL IR created during the analysis in order to transform it into the TL IR from which the TL text is finally generated (see Figure 1.2). The level of analysis, and therefore the degree of abstraction provided by the IR, varies depending on how related the languages involved are. Translating between "distant" languages (such as English–Japanese) requires a deep analysis (syntactic and semantic), while the translation between related languages (for example between Romance languages) can be achieved with shallow parsing; this last type of transfer-based systems are usually referred to as *shallow-transfer* MT systems.

Hybrid approaches integrating more than one MT paradigm are receiving increasing attention. The METIS-II (Dirix et al., 2005) MT system is an example of hybridization around the EBMT framework; it avoids the usual need for parallel corpora by using a bilingual dictionary (similar to that found in most RBMT systems) and a monolingual corpus in the TL. An example of hybridization around the rule-based paradigm is given by Oepen et al. (2007); they integrate statistical methods within an RBMT system to choose the best translation from a set of competing hypotheses (translations) generated using rule-based methods. In SMT, Koehn and Hoang (2007) integrate additional annotations at the word-level into the translation models in order to better learn some aspects of the translation that are best explained on a morphological, syntactic or semantic level. Another hybridization around the statistical approach to MT is provided by Groves and Way (2005); they combine both corpus-based methods into a single MT system by using *phrases* (sub-sentential chunks) both from EBMT and SMT into an SMT system. A different hybridization happens when an RBMT system and an SMT system are used in a cascade; Simard et al. (2007) propose an approach, analogous to that by Dugast et al. (2007), that consists of using an SMT system as an automatic post-editor of the translations produced by an RBMT system.

1.3 Rule-based machine translation

Although during the last few years the growing availability of machine-readable monolingual and parallel corpora has caused corpus-based approaches to become increasingly interesting, RBMT systems are still being actively developed mainly because:

1. corpus-based MT systems require large amounts, in the order of tens of millions of words, of parallel corpora to achieve a reasonable translation quality (Och, 2005) in open-domain tasks. Such a vast amount of parallel corpora is not available for most less-resourced language pairs demanding MT services (Forcada, 2006), such as Occitan–Catalan, French–Catalan or English–Afrikaans, among others; and

2. RBMT systems are easier to diagnose during development and the translation errors they produce have a repetitive nature, making them more predictable and easier to post-edit, and therefore, better suited for dissemination purposes.[2]

This thesis focuses on the development of shallow-transfer MT systems between related languages. Building this kind of MT system usually involves the development of:

- *monolingual* dictionaries containing all possible levels of lexical analysis (lemma, PoS and morphological inflection information) of each word;
- disambiguation methods to solve the lexical ambiguity of the SL words that may have more than one interpretation. This involves PoS tagging (Manning and Schütze, 1999, ch. 10) and, depending on the translation system, word sense disambiguation (Ide and Véronis, 1998; Agirre and Edmonds, 2007);
- *bilingual* dictionaries that for a given lemma in the SL (and perhaps some sense information) provides its translation into the TL; and
- *structural transfer* rules that detect phrases or chunks deserving special processing for word-reordering, or gender and number agreement, among others.

The shallow-transfer MT system considered in this thesis (Apertium, see appendix A) uses only a PoS tagger to solve the lexical ambiguity of SL texts, as the bilingual dictionary provides a single equivalent in the TL for each SL lemma. This approach has been proved to be adequate for translation between closely-related language pairs such as Spanish–Catalan and Occitan–Catalan.

1.4 Problems addressed and state of the art

This dissertation focuses on the hybridization of shallow-transfer RBMT systems by using corpus-based methods to learn in an unsupervised way some of the linguistic

[2]This can be easily demonstrated by trying some simple examples through Google's translation services at `http://www.google.es/language_tools`. For instance, the translation into English of the Spanish sentence *Me lo regaló tu madre* is *I gave what your mother*, while the Spanish sentence *me lo regaló tu madre* is translated as *I gave your mother*; note that both Spanish sentences only differ in the case of the first letter; however, the translation is completely different. Incidentally, both are wrong, the correct translation is *Your mother gave me a present*. Similar examples can be found at `http://www.euromatrix.net/deliverables/deliverable61.pdf`.

resources required to build a shallow-transfer RBMT system from scratch, so that the total human effort needed is reduced. The resulting hybrid system preserves most of the advantages of the rule-based approach (easier diagnosis and more predictable errors) while reducing the need for human intervention in its development.

More precisely, the methods that will be proposed in this thesis focus on how to obtain in an unsupervised way two of the resources that are required by shallow-transfer MT systems: (i) the PoS tagger used to solve the PoS ambiguities of the SL texts to translate, and (ii) the set of shallow-transfer rules that are used to convert an SL IR into the TL IR from which the TL text will be generated.

1.4.1 Part-of-speech tagging for machine translation

PoS tagging is a well-known problem and a common step in many natural-language processing applications. A PoS tagger is a program that attempts to assign the correct PoS tag or *lexical category* to all words of a given text, typically by relying on the assumption that a word can be assigned a single PoS tag by looking at the PoS tags of neighboring words.

Usually PoS tags are assigned to words by looking them up in a lexicon, or by using a morphological analyzer (Merialdo, 1994). A large portion of the words found in a text have only one possible PoS, but there are *ambiguous* words that have more than one PoS tag;[3] for example, the English word *book* can be either a noun (*She bought a book for you*) or a verb (*We need to book a room*).

Impact on translation quality

The choice of the correct PoS tag may be crucial when translating to another language because the translation of a word may greatly differ depending on its PoS; in the previous example, the English word *book* may be translated into Spanish as *libro* or as *reservo* depending on the PoS (noun or verb, respectively). However, not all words incorrectly tagged are wrongly translated since some of them may be involved in a free-ride phenomenon. A *free-ride* phenomenon happens when choosing the incorrect interpretation for an ambiguous word in a certain context does not cause a translation error. The more related two languages are, the more often this free-ride phenomenon may occur.

The following example illustrates the free-ride phenomenon. Consider the French noun phrase *la ville* being translated into Spanish by means of an RBMT system; the morphological analysis according to the lexicon of that phrase is: *la* (article or pronoun) and *ville* (noun). Before translating, the ambiguity of word *la* must be solved; however, even if *la* is incorrectly tagged as a pronoun the translation of the

[3] In Romance language texts about one word out of three is usually ambiguous.

Language	PoS-ambiguity	non-free PoS-ambiguity	# words
Spanish	23.0%	6.4%	10 066
Occitan	30.6%	21.7%	10 079
French	29.3%	17.1%	10 154

Table 1.1: For three different languages, percentage of ambiguous words (PoS-amb.), percentage of words with more than one translation into Catalan due to PoS ambiguities (non-free PoS-amb.), and number of words of the corpora used to calculate the second and third columns.

entire phrase is still correct, as the translation into Spanish is the same (*la ciudad*) for both interpretations (PoS tags) of word *la*. In contrast, the word *la* is not involved in a free ride if the noun phrase being translated from French into Spanish is *la voiture* (same morphological analysis), since in this case a noun-phrase agreement rule is applied when *la* is interpreted as an article, producing *el coche* as translation, while the translation when *la* is wrongly disambiguated as a pronoun is *la coche*, which is not correct.

Table 1.1 shows for three different languages (Spanish, Occitan[4] and French), the percentage of words which are ambiguous due to having more than one possible PoS (*PoS-ambiguous* words), the percentage of words that may be wrongly translated into Catalan if their PoS ambiguities are incorrectly solved (*non-free PoS-ambiguous words*) and the number of words of the corpora used to compute the given percentages.[5]

The percentage of non-free PoS-ambiguous words must be interpreted as a lower bound to the percentage of MT errors that may be produced if PoS ambiguities were incorrectly solved. It is a lower bound because the remaining PoS-ambiguous words (*free PoS-ambiguous words*) may not always 'benefit' from a free-ride phenomenon as illustrates the French-to-Spanish example *la voiture* above.

As can be seen in Table 1.1, in the case of Spanish, a large portion of the PoS-ambiguous words found are free PoS-ambiguous words. This is explained by the fact that the second most-frequent Spanish word (*la*), which accounts for 3.75% of the corpus, has two possible PoS tags, which are both translated in the same way into Catalan, except under certain circumstances due to the effect of a structural transfer rule ensuring noun-phrase agreement (again, as in the French–Spanish example above).

[4]Occitan has several dialects with strong differences between them and it does not exist as a unified language; the Occitan dialect used throughout this dissertation is known as Aranese and is spoken in the Val d'Aran, a small valley of the Pyrenees of Catalonia, where it is official (with some limitations) together with Catalan and Spanish.

[5]These corpora are the same used for evaluation purposes in the experiments reported in Chapters 2, 3 and 4. Section 2.5 will provide more details about these corpora and the data (morphological and bilingual dictionaries) used to calculate the percentages.

General-purpose part-of-speech tagging approaches

Different approaches have been followed in order to obtain robust general-purpose PoS taggers to be used in a wide variety of natural language processing applications. On the one hand, rule-based approaches either learn automatically (Brill, 1992, 1995b) or code manually rules capable of solving the PoS ambiguity. On the other hand, statistical approaches (Dermatas and Kokkinakis, 1995; Sánchez-Villamil et al., 2004) use corpora to estimate the parameters of a probability model that is then used to perform the PoS tagging of new corpora.

The classical statistical approach to PoS tagging followed in this thesis uses hidden Markov models (HMM: Cutting et al. 1992; Rabiner 1989; Baum and Petrie 1966). When HMMs are used for PoS tagging, each hidden state is made to correspond to a different PoS tag and the set of observable outputs is made to correspond to *word classes* which, in general, may be any suitable partition of the vocabulary.[6] The ambiguity is solved by assigning to each word the PoS tag represented by the corresponding state in the sequence of states that maximizes, given a set of HMM parameters previously estimated, the probability of the sequence of word classes observed. Appendix B provides a more detailed description of the use of HMM for PoS tagging.

HMMs can be trained in a *supervised* way from hand-tagged corpora via the *maximum-likelihood estimate* (MLE) method (Gale and Church, 1990). A *hand-tagged* (or just *tagged*) corpus is a text in which each PoS ambiguity has been solved by a human expert; therefore, such tagged corpora are a very expensive linguistic resource which is not always available, especially for less-resourced languages.

If a tagged corpus is not available, HMMs can be trained in an unsupervised way by using untagged corpora as input to the Baum-Welch *expectation-maximization* (EM) algorithm (Baum, 1972). An *untagged* corpus (Merialdo, 1994) is a text in which each word has been assigned the set of all possible PoS tags that it could receive independently of the context. This kind of text can be automatically obtained if a morphological analyzer or a lexicon is available, as is usual in RBMT. In an untagged corpus ambiguous words receive more than one PoS tag.

The approach in this thesis

The two methods (supervised and unsupervised) mentioned above to train HMM-based PoS taggers use only information from the language being tagged, a natural approach when PoS tagging is to be applied in natural language processing applications involving only one language. However, when PoS taggers are used in MT, that is, when tagging is viewed just as an intermediate task for the whole translation procedure, there are two points to which the research community has not paid enough attention:

[6] Word classes are often made to correspond to *ambiguity classes* (Cutting et al., 1992), that is, to the set of all possible PoS tags that a word could receive.

- on the one hand, that there is a natural source of knowledge, in addition to *parallel* corpora (Yarowsky and Ngai, 2001; Dien and Kiem, 2003), that can be used during training to get better PoS taggers: the use of a statistical model of the TL; and,

- on the other hand, that PoS tagging is in MT just an intermediate step needed to produce good translations into TL; therefore, what really matters is translation quality, rather than PoS tagging accuracy; in other words, one may not care whether a word is incorrectly tagged at a certain point if it gets translated correctly.

Chapter 2 describes a method, inspired by the two facts mentioned above, to train an HMM-based PoS tagger to be used in RBMT by using information, not only from the SL, but also from the TL and from the remaining modules of the MT system in which the obtained PoS tagger is to be embedded. This new approach is the first one, as far as I know, that focuses on the task in which the resulting PoS tagger will be applied by trying to maximize the quality of the whole task, instead of the tagging performance in isolation. Moreover, this new approach makes use of information from another language (the TL) to train an SL PoS tagger without using any parallel corpus. To benefit from this new training method RBMT developers only need to build the remaining modules of the translation engine before applying it to obtain the PoS taggers to be used in that RBMT system. Chapter 3 describes a method to make this training method much faster without degrading its performance.

1.4.2 Part-of-speech tag clustering

The first step in many RBMT systems is the morphological analysis of the SL text to be translated; such analysis consists of determining the set of possible lexical forms for each *surface form* (lexical unit as it appears in the SL text to be translated). Each of the possible *lexical forms* of a surface form consist of *lemma*, lexical category and morphological inflection information for that surface form. For instance, one of the lexical forms of the Spanish surface form *cantábamos* is *cantar*`-(verb.pret.1st.pl)`, which conveys the following information: lemma *cantar*, lexical category verb, and morphological inflection information preterite tense, first person, plural. The lexical category and morphological inflection information provided by a lexical form (in the example above `verb.pret.1st.pl`) will be referred to in this thesis as a *fine-grained* PoS tag.

If fine-grained PoS tags are directly used for disambiguation, the number of HMM states becomes considerably high, worsening the data sparseness problem,[7] and possibly, the PoS tagging performance. Note that in a first-order HMM the number of

[7]The data sparseness problem is caused when there are a high number of parameters that achieve a null-frequency count because no evidence for them can be collected from the training corpus.

parameters to estimate grows quadratically with respect to the number of states. To avoid using such a large number of tags (states), a tagset that groups fine-grained tags into coarser ones is usually manually defined following linguistic guidelines. However, the automatic inference of the tagset is also possible (see below).

Manual definition of the tagset

The tagset must be carefully designed. The main goal when defining the tagset is to use the least possible number of tags in order to have the smallest possible number of parameters to estimate. To achieve this fine-grained PoS tags are grouped into coarser ones, but avoiding grouping tags having different syntactic roles because this would result in poor tagging results. In particular, when PoS taggers are used in MT systems, what really counts is to be able to distinguish between analyses leading to different translations.

Sometimes, in order to improve accuracy, *partially lexicalized* HMMs may be useful. In partially lexicalized HMMs some word classes are chosen to hold only a single word. In this way the model can deal better with the peculiarities in the distribution of certain frequent words. There are two types of lexicalization:

- a lexicalization that only adds a new observable output (word class) holding the single word receiving a special treatment (Cutting et al., 1992); and,
- a lexicalization that, in addition to the definition of a new word class, also adds new states for those tags assigned to words receiving a specific treatment (Pla and Molina, 2004; Kim et al., 1999).

The latter is used in the manually defined tagsets used in Chapters 2 and 3.

Automatic inference of the tagset

The manual definition of the set of states to be used for HMM-based PoS tagging involves a human effort that would be desirable to avoid. Moreover, linguistically motivated tagsets do not guarantee better PoS tagging performance or translation quality because the underlying assumption, namely that fine-grained PoS tags having the same lexical category usually have similar probability distributions does not necessarily hold for all lexical categories. Furthermore, not all the information provided by fine-grained PoS tags is useful for probability estimation, and the amount of information that is useful because it allows to discriminate between different analyses may vary from one lexical category to another.

There have been some attempts to automatically define the HMM *topology*, that is, the set of hidden states (in our case PoS tags) to use. Stolcke and Omohundro (1994)

describe a technique for inducing the HMM structure from data, which is based in the general *model merging* strategy (Omohundro, 1992); their work focuses on HMMs for speech recognition and has the advantage of estimating the HMM topology and the transition and emission probabilities at the same time. The model merging method starts with a maximum-likelihood HMM that directly encodes the training data, that is, where there is exactly one path for each element (utterance) in the training corpus, and each path is used only by one element. Then, in each step two HMM states are selected for merging and combined into a single state, updating the transition and emission probabilities accordingly. The states to merge are chosen by using an error measure to compare the goodness of the various candidates for merging.

The approach by Stolcke and Omohundro (1994) is not directly applicable for the automatic definition of the tagset to be used by our PoS tagger because:

1. additional restrictions need to be taken into account to infer the topology of an HMM to be used for PoS tagging in order to prevent the loss of information (provided by the fine-grained PoS tags) as a consequence of a state merging;

2. the use of a maximum-likelihood HMM requires a hand-tagged corpus to build the initial model that encodes that corpus;

3. it is not feasible when the resulting HMM is intended to be used in a real environment, such as an MT system, in which previously unseen events might occur; and

4. the model merging strategy is a very time consuming task, in general $O(l^4)$ where l is the length of the training corpus.

Brants (1995a,b) focuses on the problem of finding the structure of an HMM used for PoS tagging. In his work, the author also follows the model merging technique to find the tagset to be used, but taking into account some restrictions in order to preserve the information provided by the fine-grained PoS tags (states) of the initial HMM, and smoothing the resulting probabilities to allow for the recognition of sequences of PoS tags not present in the training corpora. The initial model has one state per fine-grained PoS, not per word occurrence, and it is trained in a supervised way from hand-tagged corpora. In addition, some restrictions on the merging procedure are applied to find the pair of states to merge in polynomial time with a best-first search.

In a later work, Brants (1996) explores the *model splitting* strategy which, in contrast to model merging, selects an HMM state to be divided into two new states, updating the transitions and emission probabilities accordingly. The state selected for splitting is the one that maximizes the divergence between the resulting probability distributions after splitting. The exponential growth of the number of possible splittings makes the computation of the global maximum unfeasible, forcing the use of heuristics to find a local maximum.

Although the methods proposed by Brants (1995a,b, 1996) reduce the number of states, and consequently the number of probabilities to estimate, model merging, model splitting and the combination of both (Brants, 1996) use a hand-tagged corpus to estimate the initial model that encodes the training corpus, or to estimate the probability distribution of the newly created states through the model splitting method. Moreover, the methods of Brants decide which states to merge (or split) by trying to maximize the likelihood of the training corpus, therefore, trying to maximize tagging performance, not translation quality.

The approach in this thesis

Chapter 4 describes the use of a bottom-up agglomerative clustering algorithm to automatically infer the tagset to be used by PoS taggers involved in RBMT. Bottom-up agglomerative clustering has already been used for HMM state clustering in speech recognition tasks (Rivlin et al., 1997). The agglomerative clustering is performed over the states of an initial HMM trained using the fine-grained PoS tags delivered by the morphological analyzer through the method that will be described in Chapter 2; therefore, no hand-tagged corpora are needed.

The algorithm begins with as many clusters as there are fine-grained PoS tags, and in each step those clusters that are closer are merged into a single one only if an additional constraint analogous to that used by Brants (1995a,b) is met. This constraint is used to prevent the clustering algorithm from putting in the same cluster two or more fine-grained PoS tags if they can emit the same word class; therefore, the constraint is used to preserve the information provided by the fine-grained PoS tags along the clustering. Clustering stops when there are no more clusters to merge either because their distance is larger than a specified threshold, or because the constraint does not hold.

Although this approach is not very different from the one described by Brants (1995b), apart from the use of a clustering strategy instead of the model merging technique, the initial model is completely different. In the PoS tagger trained with the training method that will be discussed in Chapter 2, what really counts is MT *quality* rather than PoS tagging performance; therefore, the clustering method is expected to infer tagsets which are better suited to the purpose of translation.

1.4.3 Inference of shallow-transfer machine translation rules

In transfer-based MT, structural transfer rules are needed to perform syntactic and lexical changes in order to produce grammatically correct translations in the TL. The development of such transfer rules requires qualified people to code them manually, thus making them an expensive resource.

Approaches to the inference of transfer rules

Different approaches have been followed in order to learn automatically or semi-automatically the structural transformations needed to produce correct translations into the TL. Those approaches can be classified according to the translation framework in which the learned rules are applied.

Some approaches learn transfer rules to be used in RBMT. Probst et al. (2002) and Lavie et al. (2004) developed a method to learn transfer rules for MT involving less-resourced languages (such as Quechua) with very limited resources. To this end, a small parallel corpus (of a few thousand sentences) is built with the help of a small set of bilingual speakers of the two languages. The parallel corpus is obtained by translating a *controlled corpus* from the language with more resources (English or Spanish) to the less-resourced language by means of an elicitation tool. This controlled corpus consists of a list of sentences covering major linguistic phenomena in typologically diverse languages. The elicitation tool is also used to graphically annotate the word alignments between the two sentences. Finally, hierarchical syntactic rules, that can be seen as constituting a context-free transfer grammar, are inferred from the aligned parallel corpus and fed into the Stat-XFER MT engine (Lavie, 2008).

Caseli et al. (2006) propose a method to infer bilingual resources (transfer rules and bilingual dictionaries) to be used in shallow-transfer MT from aligned parallel corpora. Prior to the generation of transfer rules, *alignment blocks* (sequences of aligned words) are built from the translation examples found in the parallel corpus by considering the type of the alignments between the words. Then, shallow-transfer rules are built in a three-step procedure. In the first step, they identify the patterns in two phases, monolingual and bilingual; then in a second step their method generates shallow-transfer rules by deriving monolingual and bilingual constraints that can also be seen as the rule itself; and finally, in a third step the rules are filtered in order to solve the ambiguity caused by rules matching the same SL sequence of words. An interesting property of the inferred rules is that they are human-readable and may therefore be post-edited by human experts to improve their performance.

In the EBMT framework, some researchers have dealt with the problem of inferring a kind of translation rule called translation templates (Kaji et al., 1992; Brown, 1999; Cicekli and Güvenir, 2001). A *translation template* can be defined as a bilingual pair of sentences in which corresponding units (words or phrases) are coupled and replaced by variables. Liu and Zong (2004) provide an interesting review of the different research dealing with translation templates. Brown (1999) uses a parallel corpus and some linguistic knowledge in the form of equivalence classes (both syntactic and semantic) to perform a generalization over the bilingual examples collected. The method works by replacing each word by its corresponding equivalence class and then using a set of grammar rules to replace patterns of words and tokens by more general tokens. Cicekli and Güvenir (2001) formulate the acquisition of translation templates as a machine learning problem, in which the translation templates are learned from the

differences and similarities observed in a set of different translation examples, using no morphological information at all. Kaji et al. (1992) use a bilingual dictionary and a syntactic parser to determine the correspondences between translation units while learning the translation templates.

In the SMT framework the use of alignment templates (AT: Och and Ney (2004)) can be seen as an integration of translation rules into statistical translation models, since an AT is a generalization of the transformations to apply when translating SL into TL by using word classes.

The approach in this thesis

Chapter 5 describes an unsupervised method to infer shallow-transfer rules from parallel corpora to be used in MT. The inferred transfer rules are based on ATs, like those used in SMT. To adapt the AT approach to the RBMT framework ATs are extended with a set of restrictions to control their application as structural shallow-transfer rules.

This approach differs from those applied in the EBMT framework (see above in this section) because, on the one hand, the transfer rules generated through the method proposed in Chapter 5 are mainly based on lexical forms and, on the other hand, because they are flatter, less structured and non-hierarchical, which makes them suitable for shallow-transfer MT. Moreover, the way in which translation rules are chosen for application differs greatly from those used in the EBMT framework.

The approach in Chapter 5 also differs from the approach by Caseli et al. (2006) in how the rules are induced; while the approach in Chapter 5 uses *bilingual phrase pairs* without worrying about the type of alignments between the words, the way in which Caseli et al. (2006) induce rules depends on the type of the alignment blocks. In addition the approach in Chapter 5 does not produce rules matching the same sequence of SL items, and so no ambiguity needs to be solved.

Like the rules by Caseli et al. (2006), the shallow-transfer rules inferred by adapting the AT approach to the RBMT paradigm are human-readable, which allow human experts to edit the inferred rules so as to improve them or introduce new ones. MT developers can use this method to infer an initial set of rules and then improve them by focusing on the more difficult issues.

Chapter 2

Part-of-speech tagging for machine translation

This chapter describes a new unsupervised training method aimed at producing PoS taggers to be used in RBMT. The method uses information, on the one hand from the TL, and on the other hand, from the remaining modules of the RBMT system in which the resulting PoS tagger is to be embedded, to train an HMM-based PoS tagger for the SL. The experimental results demonstrate that, when the PoS taggers are intended to be used for MT, information from the TL can be used in the training phase to increase the translation quality of the whole MT system. The translation quality of the MT system embedding a PoS tagger trained in an unsupervised manner through this new method is clearly better than that of the same MT system embedding a PoS tagger trained through the (unsupervised) Baum-Welch EM algorithm; furthermore, the translation quality achieved is comparable to that obtained by embedding a PoS tagger trained in a supervised way from hand-tagged corpora.

2.1 Using target-language information to train part-of-speech taggers

This chapter presents a method to ease the development of an RBMT system by avoiding having to manually disambiguate an SL text to train the HMM-based PoS tagger of that MT system. The method uses, in an unsupervised way, some of the modules of the RBMT system in which the PoS tagger will be integrated, and a source of knowledge (the TL) that is readily available when PoS tagging is used for MT.[1]

[1]Although the reader may think that this new method needs an RBMT system to exist, it is actually the other way around: developers building an RBMT system may use this new method to build the PoS tagger of that MT system in an unsupervised way.

The main idea behind the use of TL information is that the correct disambiguation (tag assignment) of a given SL segment will produce a more likely TL translation than any (or most) of the remaining wrong disambiguations. As the resulting SL PoS tagger is intended to be used in MT, attention must be focused on MT performance rather than on PoS tagging accuracy.

2.1.1 Background

Yarowsky and Ngai (2001) proposed a method which also uses information from TL in order to train PoS taggers. They considered, however, information from aligned parallel corpora and from (at least) one manually tagged corpus for the TL. A similar approach is followed by Dien and Kiem (2003) who use a Vietnamese–English parallel corpus and the transformation-based learning (TBL) method (Brill, 1995a) to bootstrap a PoS-annotated English corpus by exploiting the PoS information of the corresponding Vietnamese words. Then, they project the PoS annotations from the English side of the parallel corpus to the Vietnamese side through the word alignments. Finally they manually correct the resulting Vietnamese PoS-annotated corpus. In contrast, the method described in this chapter needs neither aligned parallel corpora nor manually tagged texts. Moreover, the method views PoS tagging as an intermediate task for the translation procedure, instead of as an objective in its own right.

Carbonell et al. (2006) proposed a new MT framework in which a large full-form bilingual dictionary (containing inflected words and their equivalents in the TL) and a huge TL corpus is used to carry out the translation; neither parallel corpora nor transfer rules are needed. The idea behind Carbonell's paper and the one in this chapter share the same principle: if the goal is to get good translations into TL, let a model of the TL decide whether a given "construction" in the TL is good or not. In contrast, Carbonell's method uses TL information at translation time, while the approach in this chapter uses only TL information when training one module that is then used, in conjunction with the rest of the MT modules, to carry out the translation; therefore, no TL information is used at all by the approach in this chapter at translation time, which makes the whole MT system much faster.

2.1.2 Overview of the method

This new method works as follows:

- For a given segment (word sequence) s in the SL, all possible disambiguation choices (combinations of the PoS tags for each word) $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_N$ are considered;[2]

[2]Each SL segment s is analyzed using the morphological analyzer of the MT system; for each SL word the set of possible PoS tags is obtained.

- the SL segment s is translated into the TL according to each possible disambiguation $\boldsymbol{g}$ by using the modules of the MT system subsequent to PoS tagging (see Figure A.1 on page 109);

- each of the resulting translations $\tau(\boldsymbol{g}, s)$ is scored against a probabilistic TL model M_{TL};

- the probability $P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))$ of each translation $\tau(\boldsymbol{g}, s)$ in the language model M_{TL} is used to estimate the probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ of each disambiguation $\boldsymbol{g}$ given the SL segment s in the tagging model M_{tag} we are trying to learn; and, finally,

- the estimated probabilities $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ are used to determine the parameters of the tagging model M_{tag} by using them as partial counts, that is, as if disambiguation $\boldsymbol{g}$ had been seen $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ times in the training corpus for the SL segment s.

The following example illustrates how the method works. Suppose that we are training an English PoS tagger to be used within an RBMT system which translates from English into Spanish, and that we have the following segment in English, $s =$ *"He rocks the table"*. The first step is to use the morphological analyzer of the MT system to obtain the set of all possible PoS tags for each word. Suppose that the morphological analysis of the previous segment according to the lexicon is: *He* (pronoun), *rocks* (verb or noun), *the* (article) and *table* (verb or noun). As there are two ambiguous words with two possible PoS tags each (*rocks* and *table*) there are, for the given segment, four disambiguation choices or PoS combinations:

- $\boldsymbol{g}_1 =$ (pronoun, verb, article, noun),

- $\boldsymbol{g}_2 =$ (pronoun, verb, article, verb),

- $\boldsymbol{g}_3 =$ (pronoun, noun, article, noun), and

- $\boldsymbol{g}_4 =$ (pronoun, noun, article, verb).

The next step is to translate the English (SL) segment into Spanish (TL) according to each disambiguation $\boldsymbol{g}$:

- $\tau(\boldsymbol{g}_1, s) =$ "*Él mece la mesa*",

- $\tau(\boldsymbol{g}_2, s) =$"*Él mece la presenta*",

- $\tau(\boldsymbol{g}_3, s) =$"*Él rocas la mesa*", and

- $\tau(\boldsymbol{g}_4, s) =$"*Él rocas la presenta*".

Then each translation $\tau(\boldsymbol{g}, s)$ is scored against a Spanish language model M_{TL}. It is expected that a reasonable Spanish language model M_{TL} will give a higher likelihood $P_{\mathrm{TL}}(\tau(\boldsymbol{g}_1, s))$ to $\tau(\boldsymbol{g}_1, s)$ than to the remaining translations ($\tau(\boldsymbol{g}_2, s)$, $\tau(\boldsymbol{g}_3, s)$ and $\tau(\boldsymbol{g}_4, s)$), as they make little sense in Spanish. In the method presented here, the probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ of each tag sequence $\boldsymbol{g}$ given the SL segment s in the tagging model M_{tag} is taken to be proportional to the likelihood $P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))$ of their respective translations into the TL, and then used to estimate the HMM parameters.

As the previous example illustrates, the method uses an untagged SL corpus as input, the remaining modules of the RBMT system following the PoS tagger, and a TL model M_{TL}. The input SL corpus must be *segmented* before training in order to consider all disambiguations for each segment independently from the others. In this work, a *segment* is a sequence of words that is processed independently from the adjacent segments by the MT modules following the PoS tagger. Concerning the TL model, a language model based on trigrams of words has been considered. Section 2.5 reports the results of experiments in which PoS taggers to translate from Spanish, Occitan and French into Catalan are trained; therefore, a Catalan language model is used.

2.2 HMM training for machine translation

This section presents the mathematical details of the unsupervised method to train SL HMM-based PoS taggers to be used in MT introduced in the previous section; as the goal is to train PoS taggers for their use in MT, this new training method will be referred as an *MT-oriented* method. Despite the fact that information in the rest of the modules of the MT system is used, this training method may be said to be unsupervised because no hand-tagged corpora are needed.

Learning an SL PoS tagging model M_{tag} using information from both TL and SL by means of an MT system can be seen as trying to approximate the following equation:

$$P_{\mathrm{TL}}(t) \simeq P_{\mathrm{trans,tag,SL}}(t), \tag{2.1}$$

that is, approximating the probability $P_{\mathrm{TL}}(t)$ of every TL segment t in a TL model M_{TL} as the probability of t in a composite model consisting of a translation model M_{trans}, the PoS tagger model M_{tag} whose parameters we are trying to learn, and an SL model M_{SL}.

As the goal is to learn the parameters of the SL PoS tagger model M_{tag}, special attention must be paid to all possible disambiguations (PoS combinations). Taking this into account, and the way in which a TL segment t would be produced from an SL segment s through an MT model M_{trans} and an SL PoS tagger M_{tag}, the right side

of Equation (2.1) can be rewritten as:

$$P_{\text{trans,tag,SL}}(t) = \sum_{s} \sum_{\boldsymbol{g}} P_{\text{trans}}(t|\boldsymbol{g}, s)\ P_{\text{tag}}(\boldsymbol{g}|s)\ P_{\text{SL}}(s), \tag{2.2}$$

where $\boldsymbol{g} = (\gamma_1 \dots \gamma_N)$ is a sequence of PoS tags in the SL; $P_{\text{trans}}(t|\boldsymbol{g}, s)$ is the probability in the translation model M_{trans} of a TL segment t given a tag sequence $\boldsymbol{g}$ and an SL segment s; $P_{\text{tag}}(\boldsymbol{g}|s)$ is the probability in the PoS tagger model M_{tag} of the tag sequence $\boldsymbol{g}$ given the source segment s; and $P_{\text{SL}}(s)$ is the probability in the SL model M_{SL} of that SL segment. The unrestricted sums over all possible s and over all possible tag sequences $\boldsymbol{g}$ are in principle necessary unless we know more about the models.

Once we have the general equation describing how the tagging model M_{tag} is related to the TL within an MT system, some choices regarding the models being used can be made. The translation model chosen is an RBMT system which assigns a single TL segment $\tau(\boldsymbol{g}, s)$ to each source segment s and PoS tag sequence $\boldsymbol{g}$; therefore, we can write:

$$P_{\text{trans}}(t|\boldsymbol{g}, s) = \delta_{t,\tau(\boldsymbol{g},s)}, \tag{2.3}$$

where $\delta_{a,b}$ is the Kronecker delta ($\delta_{a,b} = 1$ if $a = b$ and zero otherwise).[3] Thus, the basic equation can be then rewritten to integrate the translation model as follows:

$$P_{\text{trans,tag,SL}}(t) = \sum_{s} \sum_{\boldsymbol{g}} \delta_{t,\tau(\boldsymbol{g},s)} P_{\text{tag}}(\boldsymbol{g}|s)\ P_{\text{SL}}(s). \tag{2.4}$$

The PoS tagging model M_{tag} has been chosen to be an HMM $\lambda = (\Gamma, \Sigma, A, B, \pi)$, in where Γ refers to the set of hidden states (PoS tags), Σ refers to the set of observable outputs (word classes), and A, B and π to the transition probabilities, emission probabilities and initial probabilities, respectively. Appendix B gives an extensive explanation of how HMMs are used to perform PoS tagging and the assumptions made to avoid learning the probability π of each PoS tag being the initial one.

As a consequence of the tagging model M_{tag}, the set $T(s)$ of PoS tag sequences $\boldsymbol{g}$ that can be assigned to a source segment s is finite, and equal to all possible PoS tag combinations of words in s. Because of this we will call each $\boldsymbol{g}$ a (disambiguation) *path* as it describes an unique state path in the HMM.

At this point, Equation (2.4) can be rewritten as follows:

$$P_{\text{trans,tag,SL}}(t) = \sum_{s:\ \tau(\boldsymbol{g},s)=t,\ \boldsymbol{g}\in T(s)} P_{\text{tag}}(\boldsymbol{g}|s)\ P_{\text{SL}}(s), \tag{2.5}$$

where the translation model has been integrated as a restriction over summations.

[3] A different model could be used; for instance, one where a segment s tagged as $\boldsymbol{g}$ could have more than one translation ("polysemy"), that is, one where $\tau(\boldsymbol{g}, s)$ is a set. This model would have additional parameters that would have to be known or trained, or else, additional approximations would have to be made.

Now that the particular PoS tagging model to be learned (M_{tag}) and the translation model to be used (M_{trans}) have been integrated, some approximations and assumptions need to be made in order for the method to work in a practical framework.

Approximations and assumptions. As the translation model M_{trans} has no analytical form and the number of possible s is in principle infinite, it is unfeasible to solve Equation (2.5) for all possible $P_{\text{tag}}(\boldsymbol{g}|s)$, even if an SL model M_{SL} is available. However, we are not interested in values of $P_{\text{tag}}(\boldsymbol{g}|s)$ but instead in training the parameters A and B of an approximate model that computes them. Therefore, the method will take *samples* from an SL model made of representative SL texts; that is, a representative SL corpus will be used as a *source* of segments to process (approximation #1). An additional approximation here is as follows: when computing the contribution of each segment s to the HMM parameters A and B, the possible contributions of other SL segments s' to the same translation t may be safely *ignored* (approximation #2); that is, it is assumed that it is unlikely that a segment s' has the same translation that s has for some disambiguation $\boldsymbol{g}'$.

Applying it to a single sampled segment s, Equation (2.5) may be written as:

$$P_{\text{trans,tag}}(t|s) = \sum_{\tau(\boldsymbol{g},s)=t,\ \boldsymbol{g}\in T(s)} P_{\text{tag}}(\boldsymbol{g}|s), \tag{2.6}$$

where the probability of a target language segment t given the SL segment s is computed as the sum over all disambiguations $\boldsymbol{g} \in T(s)$ of the probability of each $\boldsymbol{g}$ given the source segment s and the HMM M_{tag} we are trying to learn.

The main assumption in this work is that the probability $P_{\text{trans,tag}}(t|s)$ can be approximated through a TL model as follows (approximation #3):

$$P_{\text{trans,tag}}(t|s) \simeq \begin{cases} \frac{1}{k_s}\, P_{\text{TL}}(t) & \text{if } \exists \boldsymbol{g}:\ \tau(\boldsymbol{g},s)=t \\ 0 & \text{otherwise} \end{cases}, \tag{2.7}$$

with

$$k_s = \sum_{t':\ (\exists \boldsymbol{g}:\ \tau(\boldsymbol{g},s)=t')} P_{\text{TL}}(t'), \tag{2.8}$$

where k_s is the sum of the probabilities of all possible translations into TL of the SL segment s according to all the disambiguations given by $T(s)$; that is, the probabilities of TL sentences t_i that cannot be produced as a result of the translation of SL segment s by means of the MT system being used are not taken into account.

At this point there are two different ways of computing the probability $P_{\text{trans,tag}}(t|s)$. Making both right-hand sides of Equations (2.6) and (2.7) equal when $\tau(\boldsymbol{g}, s) = t$ yields:

$$\sum_{\substack{\boldsymbol{g}'\in T(s),\\ \tau(\boldsymbol{g}',s)=\tau(\boldsymbol{g},s)}} P_{\text{tag}}(\boldsymbol{g}'|s) \simeq \frac{1}{k_s}\, P_{\text{TL}}(\tau(\boldsymbol{g}, s)), \tag{2.9}$$

where t has been replaced by $\tau(\boldsymbol{g}, s)$ because of the restriction over t introduced by the translation model M_{trans}. From now on $\tau(\boldsymbol{g}, s)$ will be used instead of t to mean that this restriction holds.

Note that Equation (2.9) takes into account the free-ride phenomenon already described in Section 1.4.1, as more than one $\boldsymbol{g}$ may contribute to the same translation $\tau(\boldsymbol{g}, s)$. Let $\xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s)$ be a factor that measures the (fractional) contribution of disambiguation $\boldsymbol{g}$ to the translation into TL $\tau(\boldsymbol{g}, s)$ of segment s, that is, $\xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s)$ dictates how the probability $P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))$ must be shared out, after normalization, between all the disambiguation paths of segment s producing $\tau(\boldsymbol{g}, s)$. At this point Equation (2.9) can be rewritten as:

$$P_{\mathrm{tag}}(\boldsymbol{g}|s) \simeq \frac{1}{k_s}\, P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))\, \xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s). \tag{2.10}$$

The fact that more than one path in segment s, say $\boldsymbol{g}$ and $\boldsymbol{g}'$, produce the same translation $\tau(\boldsymbol{g}, s)$ does not necessarily imply that $\xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s) = \xi(\boldsymbol{g}', \tau(\boldsymbol{g}, s), s)$. However, in the absence of further information, the contributions of each path will be approximated as being equal (approximation #4):

$$\xi(\boldsymbol{g}, \tau(\boldsymbol{g}, s), s) \approx \frac{1}{|\{\boldsymbol{g}' \in T(s) : \tau(\boldsymbol{g}', s) = \tau(\boldsymbol{g}, s)\}|}. \tag{2.11}$$

Although this approximation may affect PoS tagging performance, it is expected to affect translation quality only indirectly; recall that the method is aimed at training PoS taggers to be used in MT; therefore, what really matters is translation quality, not tagging accuracy.

Integrating Equation (2.11) into Equation (2.10) we have:

$$P_{\mathrm{tag}}(\boldsymbol{g}|s) \simeq \frac{1}{k_s}\, \frac{P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))}{|\{\boldsymbol{g}' \in T(s) : \tau(\boldsymbol{g}', s) = \tau(\boldsymbol{g}, s)\}|}; \tag{2.12}$$

which expresses a proper probability model as can be easily shown by summing over all possible disambiguation paths $\boldsymbol{g}$ of SL segment s.

Equation (2.12) shows how a given disambiguation $\boldsymbol{g}$ of words in an SL segment s is related to the TL. Thus the values of $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ approximated in this way can be used as a fractional count to estimate the HMM parameters A and B in order to make Equation (2.12) hold as closely as possible.

The objective is to estimate the frequency counts $n(\cdot)$ needed to estimate the HMM parameters by using the approximate probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ as an information source. Then, these frequency counts can be used to calculate the HMM parameters as with any other training method through the general Equations (B.1) and (B.5) shown in appendix B (see page 115 et seq.).

Figure 2.1 summarizes the entire process followed to estimate the frequency counts. First of all, the input SL text is segmented, and all possible disambiguation paths $\boldsymbol{g}$ for

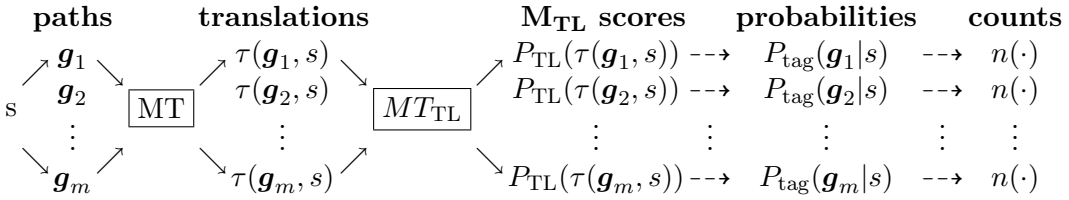


Figure 2.1: Scheme of the process followed by the MT-oriented method to estimate the frequency counts $n(\cdot)$ used to calculate the HMM parameters. These frequency counts are based on the probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ of each disambiguation $\boldsymbol{g}$ in the tagging model M_{tag} given the SL segment s.

each segment are considered. Therefore, for a given segment s the translations $\tau(\boldsymbol{g}, s)$ of segment s according to each possible disambiguation $\boldsymbol{g} \in T(s)$ are performed. Once all different translations of segment s have been obtained, each translation $\tau(\boldsymbol{g}, s)$ is scored using the TL model M_{TL}. Then these scores are used to estimate the probability $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ in the tagging model M_{tag} of each path $\boldsymbol{g}$ being the correct disambiguation of segment s using Equation (2.12). Finally, these probabilities are used to estimate the frequency counts already mentioned as described below.

The frequency counts $n(\cdot)$ used in Equations (B.1) and (B.5) are obtained from the estimated probabilities $P_{\mathrm{tag}}(\boldsymbol{g}|s)$; in this case, frequency counts are approximations $\tilde{n}(\cdot)$, instead of exact values $n(\cdot)$. In order to approximate these counts, each $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ is treated as a fractional count; that is, as if the disambiguation $\boldsymbol{g}$ of segment s had been seen $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ times. An estimate of tag occurrences based on $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ is:[4]

$$\tilde{n}(\gamma_i) \cong \sum_{n=1}^{N_S} \sum_{\boldsymbol{g} \in T(s_n)} C_{s_n,\boldsymbol{g}}(\gamma_i)\, P_{\mathrm{tag}}(\boldsymbol{g}|s_n), \tag{2.13}$$

where N_S is the number of segments in the SL training corpus, and $C_{s_n,\boldsymbol{g}}(\gamma_i)$ is the number of times tag γ_i appears in path $\boldsymbol{g}$ of segment s_n. Analogously, an estimate of the tag pair occurrence frequency based on $P_{\mathrm{tag}}(\boldsymbol{g}|s)$ is:

$$\begin{aligned} \tilde{n}(\gamma_i\gamma_j) \cong & \sum_{n=1}^{N_S} \sum_{\boldsymbol{g} \in T(s_n)} C_{s_n,\boldsymbol{g}}(\gamma_i, \gamma_j)\, P_{\mathrm{tag}}(\boldsymbol{g}|s_n) + \\ + & \sum_{n=1}^{N_S} \left(\sum_{\substack{\boldsymbol{g}' \in T(s_{n-1}),\\ \gamma_i = \mathrm{last}(\boldsymbol{g}')}} P_{\mathrm{tag}}(\boldsymbol{g}'|s_{n-1}) \sum_{\substack{\boldsymbol{g} \in T(s_n),\\ \gamma_j = \mathrm{first}(\boldsymbol{g})}} P_{\mathrm{tag}}(\boldsymbol{g}|s_n) \right), \end{aligned} \tag{2.14}$$

where $C_{s_n,\boldsymbol{g}}(\gamma_i, \gamma_j)$ is the number of times tag γ_i is followed by tag γ_j in path $\boldsymbol{g}$ of segment s_n, and $\mathrm{first}(\boldsymbol{g})$ and $\mathrm{last}(\boldsymbol{g})$ are two functions returning the first and last tag,

[4]Note that in Equations (B.1) and (B.5) s_i and v_k are used to refer to the HMM state γ_i and the word class σ_k, respectively.

$s \equiv$	*He*	*rocks*	*the*	*table*	
	$\{$ PRN $\}$	$\left\{ \begin{array}{c} \text{VB} \\ \text{NN} \end{array} \right\}$	$\{$ ART $\}$	$\left\{ \begin{array}{c} \text{VB} \\ \text{NN} \end{array} \right\}$	
					$P_{\text{tag}}(\boldsymbol{g}\|s)$
$\boldsymbol{g}_1 \equiv$	PRN	VB	ART	NN	
$\tau(\boldsymbol{g}_1, s) \equiv$	*Él*	*mece*	*la*	*mesa*	0.75
$\boldsymbol{g}_2 \equiv$	PRN	VB	ART	VB	
$\tau(\boldsymbol{g}_2, s) \equiv$	*Él*	*mece*	*la*	*presenta*	0.15
$\boldsymbol{g}_3 \equiv$	PRN	NN	ART	NN	
$\tau(\boldsymbol{g}_3, s) \equiv$	*Él*	*rocas*	*la*	*mesa*	0.06
$\boldsymbol{g}_4 \equiv$	PRN	NN	ART	VB	
$\tau(\boldsymbol{g}_4, s) \equiv$	*Él*	*rocas*	*la*	*presenta*	0.04

Figure 2.2: Example of an ambiguous SL (English) segment s, paths and translations $\tau(\boldsymbol{g}, s)$ into TL (Spanish) resulting from each possible disambiguation $\boldsymbol{g}$, and estimated probability $P_{\text{tag}}(\boldsymbol{g}|s)$ of each path being the correct disambiguation.

respectively, of a disambiguation path $\boldsymbol{g}$. Note that the second term of the addition considers the boundary between two adjacent segments.

The number of times a word class σ_k is emitted by a given tag γ_j is approximated as

$$\tilde{n}(\sigma_k, \gamma_j) \cong \sum_{n=1}^{N_S} \sum_{\boldsymbol{g} \in T(s_n)} C_{s_n,\boldsymbol{g}}(\sigma_k, \gamma_j)\, P_{\text{tag}}(\boldsymbol{g}|s_n), \tag{2.15}$$

where $C_{s_n,\boldsymbol{g}}(\sigma_k, \gamma_j)$ is the number of times word class σ_k is emitted by tag γ_j in path $\boldsymbol{g}$ of segment s_n.

Figure 2.2 shows an example of the application of the method to an isolated segment when a TL language model M_{TL} based on trigrams of words is used (see Section 2.4 for more information about the TL model used). The SL segment *He rocks the table* has four possible disambiguation paths $(\boldsymbol{g}_1, \cdots, \boldsymbol{g}_4)$ which are translated to the TL and scored using the TL model M_{TL}; then, these scores are used to estimate the probability $P_{\text{tag}}(\boldsymbol{g}|s)$ of each possible disambiguation $\boldsymbol{g}$ being the correct one for the given SL segment.

2.3 Segmenting the SL text

Previous sections have discussed segments as the SL units to be processed by the method. In Section 2.1.2 a segment was defined as a sequence of words that is processed independently of the adjacent segments by the remaining modules of the MT system after the PoS tagger. SL text segmentation must indeed be carefully designed so that two words which are jointly treated at some stage of the MT process following the

PoS tagging component are not placed in different segments. This would result in incorrect sequences in the TL (for example, if two words involved in a word reordering or agreement rule are assigned to different segments) and, as a consequence of that, in wrong likelihood estimations. However, it must be noticed that, when related languages are involved, if no transfer patterns are taken into account, a great proportion of segment translations may be still correct because of the small grammatical divergences between the languages involved (see Section 2.5.4).

Using whole sentences as segments would seem a reasonable choice, because most current MT systems perform the translation at sentence level, translating each sentence independently. However, since the number of disambiguations grows exponentially with sentence length, sentences need to be segmented so that the problem is made computationally feasible. In general, first-order HMMs can be trained by breaking the corpus into segments whose first and last words are unambiguous, since unambiguous words reveal or *unhide* the hidden state of the HMM (Cutting et al., 1992, sect. 3.4).[5] Adequate strategies for ensuring segment independence depend on the particular translation system. In Sections 2.5.3 and 2.5.4, the strategy used in each experiment will be described.

2.4 Target-language model

In previous sections we have seen that each translation $\tau(\boldsymbol{g}, s)$ must be scored against a target-language model M_{TL} to obtain the likelihoods $P_{\mathrm{TL}}(\tau(\boldsymbol{g}, s))$ to be used in Equation (2.12). In principle, there are no restrictions on the TL model to be used. This section describes the language model used in the experiments, namely a trigram model based on words as they appear in raw texts.

Trigram model of surface forms

A classical trigram model of TL surface forms is considered in this section. The trigram model can easily be trained from a raw TL corpus. Before training, the corpus must undergo a small amount of preprocessing, e.g. inserting blank spaces before and after punctuation marks, replacing each apostrophe by a space, replacing all numbers by a common identifier and switching all words to lower case. Translations produced by the system must also be preprocessed in the same way before being evaluated via this language model.

In order to prevent the model from assigning a null probability to every text segment containing an unseen trigram, probabilities are smoothed via a form of *deleted interpolation* (Jelinek, 1997, ch. 4) in which weighted estimates are taken from trigram and bigram probabilities, and a uniform probability distribution.

[5] In n-th order HMMs, segments would be delimited by n contiguous unambiguous words.

The smoothed trigram probabilities consist of a linear combination of trigram and bigram probabilities:

$$P^{(3)}_{\mathrm{TL}}(w_3|w_1,w_2) = \lambda_3(w_1,w_2)f(w_3|w_1,w_2) + (1-\lambda_3(w_1,w_2))P^{(2)}_{\mathrm{TL}}(w_3|w_2) \qquad (2.16)$$

where $\lambda_3(w_1,w_2)$ is the smoothing coefficient for trigrams, and $f(w_3|w_1,w_2)$ is the observed frequency for the trigram w_1,w_2,w_3. Bigram probabilities are smoothed in a similar manner:

$$P^{(2)}_{\mathrm{TL}}(w_3|w_2) = \lambda_2(w_2)f(w_3|w_2) + (1-\lambda_2(w_2))P^{(1)}_{\mathrm{TL}}(w_3) \qquad (2.17)$$

where $\lambda_2(w_2)$ is the smoothing coefficient for bigrams, $f(w_3|w_2)$ the observed frequency for bigram w_2w_3, and $P^{(1)}_{\mathrm{TL}}(w_3)$ the probability of having seen the word w_3.

Jelinek (1997) computes the values of the smoothing coefficients $\lambda(\cdot)$ by splitting the training corpus into the *kept* part, the larger one from which frequency counts are collected, and the *held-out* part, used to collect more counts and estimate the value of the smoothing coefficients.

An approximate way to estimate the values of the smoothing coefficients, and the one used in this thesis, is the *successive linear abstraction* method proposed by Brants and Samuelsson (1995):

$$\lambda_3(w_1,w_2) = \frac{\sqrt{n(w_1,w_2)}}{1+\sqrt{n(w_1,w_2)}}$$

$$\lambda_2(w_2) = \frac{\sqrt{n(w_2)}}{1+\sqrt{n(w_2)}}$$

where $n(w_1,w_2)$ and $n(w_2)$ are the number of occurrences of the bigram w_1,w_2 and the word w_2, respectively, in the training corpus.

Nevertheless, in spite of the smoothing techniques used, when a trigram ending in a previously unseen word is present, the final probability is still zero because the unigram probability $P^{(1)}_{\mathrm{TL}}(w_3)$ is null. To avoid this problem, unigram probabilities are smoothed as well using the Good-Turing method (Gale and Sampson, 1995). This method estimates the probabilities for the seen words given their frequencies and the joint probability of all unobserved words. Then, when the likelihood of a given input string is evaluated, the probability of an unseen word is made equal to the probability of those words that have been seen only once, as the probability of an isolated unseen word cannot be computed.

Finally, note that computing likelihoods as products of trigram probabilities causes (as in most statistical MT approaches) shorter translations of the same segment to receive higher scores than larger ones; this may have an effect on the overall performance of the HMM training strategy.

Likelihood evaluation

When estimating path likelihoods, if text segmentation is correctly performed so that segments are independent (as already mentioned), a good estimate of trigram probabilities for the translation produced by a given path can be performed independently of the context (remaining text) in which it could appear. Equation (2.18) shows the likelihood for the translation $\tau(\boldsymbol{g}, s) = w_F \ldots w_L$ independently of the context when $L > F + 1$:

$$P_{\mathrm{TL}}(w_F \ldots w_L) = P^{(1)}_{\mathrm{TL}}(w_F) P^{(2)}_{\mathrm{TL}}(w_{F+1}|w_F) \prod_{j=F+2}^{L} P^{(3)}_{\mathrm{TL}}(w_j|w_{j-2}w_{j-1}), \tag{2.18}$$

if $L = F$ the likelihood equals the probability $P^{(1)}_{\mathrm{TL}}(w_F)$ of the unigram w_F; analogously, if $L = F + 1$ the likelihood equals the product of the probability of the unigram w_F and the bigram $w_F w_{F+1}$: $P^{(1)}_{\mathrm{TL}}(w_F) P^{(2)}_{\mathrm{TL}}(w_{F+1}|w_F)$.

2.5 Experiments

2.5.1 Task and evaluation

Task

A large number of experiments have been conducted to test the MT-oriented training method. Experiments focus on three languages —Spanish, French and Occitan— all being translated into Catalan by means of the open-source shallow transfer MT platform Apertium (Armentano-Oller et al., 2006; Corbí-Bellot et al., 2005) which is described in detail in appendix A. More precisely, the publicly available language-pair packages `apertium-es-ca-1.0.2`, `apertium-fr-ca-0.9`, and `apertium-oc-ca-1.0.2` have been used to test the MT-oriented approach for the Spanish–Catalan, French–Catalan, and Occitan–Catalan (Armentano-Oller and Forcada, 2006) language pairs, respectively. Notice that for training, the whole MT engine, except the PoS tagger, is used to produce all the translations $\tau(\boldsymbol{g}, s)$ that are evaluated via the TL model M_{TL}.

Two different sets of experiments have been conducted on each language pair, all using a Catalan language model based on trigrams of words (see Section 2.4); one that uses a complete structural transfer module to produce all the translations $\tau(\boldsymbol{g}, s)$, and another in which the structural transfer module of the MT engine is simplified to a minimum (context-free word-for-word) "null" structural transfer module. The Catalan trigram model used was trained from a raw-text Catalan corpus with around 2 million words.

Language	Fine-grained tags				Coarse tags			
	single-word	**multi-word**	$\|\Gamma\|$	$\|\Sigma\|$	**single-word**	**multi-word**	$\|\Gamma\|$	$\|\Sigma\|$
Spanish	377	1 739	2 116	3 061	85	14	99	291
French	320	102	422	873	72	4	76	264
Occitan	348	1 957	2 305	3 809	87	18	105	345

Table 2.1: Main data for the tagset used by the corresponding PoS tagger for each language. Each tagset consists of a set of coarse tags which group together the fine-grained PoS tags delivered by the morphological analyzer. There are single-word tags and multi-word tags. Multi-word tags are used for SL contractions and verbs with attached clitics. Grouping fine-grained PoS tags into coarser ones reduces the total number of states $|\Gamma|$ and the number of word classes $|\Sigma|$ that need to be taken into account.

Tagset. The tagset used by the corresponding PoS tagger for each language was manually defined and consists of a set of coarse tags grouping the fine-grained PoS tags delivered by the morphological analyzer (see Section 1.4.2 for further details).

Table 2.1 summarizes the main features of the three tagsets used. The number of word classes $|\Sigma|$ is also given. In these tagsets, a few very frequent ambiguous words are assigned special hidden states (Pla and Molina, 2004), and consequently special word classes; these very frequent words are assigned special hidden states as a consequence of preliminary experiments that showed that their ambiguity is better solved if they are lexicalized. In our Spanish tagset only the words *para* (preposition or verb), *que* (conjunction or relative), *como* (preposition, relative or verb), *algo* (pronoun or adverb), and *más/menos* (adverb or adjective) are assigned special hidden states in Γ; for Occitan, words *que* (conjunction or relative), *molt* (adjective or adverb), *a* (preposition or verb), and *auer* (verb), are also assigned special hidden states; for French no special hidden states are used.

Evaluation

The performance of this new HMM training method is compared in all the experiments to that of the same MT system when using different MT setups, that is, when using a PoS tagger trained via the classical methods, or even when no PoS tagger is used at all (see reference results in Section 2.5.2). In the case of Spanish the PoS tagging performance has also been evaluated; unfortunately this evaluation was not possible in the case of the other two languages because no appropriate hand-tagged corpora were available.

PoS tagging error rate. PoS tagging errors are expressed as the percentage of incorrect tags assigned to all words (including unknown words). The Spanish PoS tagging error rates are evaluated using an independent Spanish hand-tagged corpus

Lang.	# words	# sent.	PoS-amb.	non-free PoS-amb.	Unk. words
Spanish	10 066	457	23.0 %	6.4 %	4.9 %
Occitan	10 079	538	30.6 %	21.7 %	5.0 %
French	10 154	387	29.3 %	17.1 %	10.4 %

Table 2.2: Number of SL words, number of sentences, percentage of ambiguous words (PoS-amb., without considering unknown words), percentage of words with more than one translation into Catalan due to PoS ambiguities (non-free PoS-amb.), and percentage of unknown words for each corpus used to evaluate the translation performance of each MT system when embedding the SL PoS tagger being evaluated.

with around 8 000 words. In this corpus the percentage of ambiguous words according to the lexicon, including unknown words, is 27.6% (3.9% unknown, 23.7% known). Note that, when evaluating using this tagged corpus, 0.8% of the words are always taken to be incorrectly tagged since the correct PoS tag (in the evaluation corpus) is never provided by the morphological analyzer due to incomplete morphological entries in the lexicon.

Machine translation quality. As the method is aimed at training PoS taggers to be used in MT, the evaluation of the translation performance becomes the most relevant. Table 2.2 shows, for the different SL corpora used for evaluation, the number of SL words, the number of sentences, the percentage of words which are ambiguous due to having more than one possible PoS (*PoS-ambiguous* words, PoS-amb.), the percentage of words with more than one translation into Catalan because of PoS ambiguities (*non-free PoS-ambiguous* words, non-free PoS-amb.), and the percentage of words in each evaluation corpus that are unknown to the system. Note that these evaluation corpora were presented in the introductory chapter (Section 1.4.1, Table 1.1) where the impact of PoS ambiguity on translation quality was discussed.

The data reported in Table 2.2 correspond to the SL corpora that are translated using the different MT setups discussed below; the translation used as a reference for the evaluation is a human-corrected (post-edited) machine translation into TL performed with the same linguistic data (lexicon, bilingual dictionary and structural transfer rules) and MT platform used for the experiments. The human-corrected (post-edited) translation is obtained by modifying the minimum amount of text so that the resulting translation is adequate.

Translation performance is evaluated using two different measures; on the one hand, the *word error rates* (WER), and on the other hand, the *bilingual evaluation understudy* (BLEU, Papineni et al. 2002). WERs are computed as the word-level edit distance (Levenshtein, 1965) between the translation being evaluated and the reference translation. The WER calculated using as reference a post-edited translation gives an idea of how much each method helps human translators in their daily work, since it

provides the percentage of words that need to be inserted, replaced or deleted to transform the MT output into an adequate translation into TL for dissemination purposes. Concerning the BLEU metric, it must be noted that as only one reference translation is used, and that reference is a human-corrected version of the same MT output, BLEU scores are higher than may initially be expected.

Confidence intervals

To test whether the method behaves the same way independently of the training corpora, in the following sections different training corpora will be used when available, and the mean and the standard deviation of the WER and the BLEU scores achieved after training with each corpora will be reported. In addition, the use of confidence intervals will allow for an easier interpretation of the translation quality measures and will permit a better comparison between them.

Confidence intervals of MT quality measures are calculated through the *bootstrap resampling* method as explained by Koehn (2004). In general, the bootstrap resampling method consists of estimating the precision of sample statistics (in our case, translation quality measures) by randomly resampling with replacement (that is, allowing repetitions) from the full set of samples (Efron and Tibshirani, 1994): in MT, sentences and their respective reference translations. This method has the property that no assumptions are made about the underlying distribution of the variable; in our case, the MT quality measure.

The calculation of the confidence intervals consists of the following steps:

1. the translation performance is evaluated a large number of times using randomly chosen sentences from the test corpus, and their counterpart sentences in the reference corpus;
2. all the calculated measures are sorted in ascending order; and
3. the top q% and the bottom q% elements are removed from that list.

After that, the remaining values are in the interval $[a, b]$. This interval approximates with probability $1 - 2q/100$ the range of values in which the quality measure being reported lies for evaluation corpora with a number of sentences equal to that used to carry out the evaluation (see Table 2.2).

2.5.2 Reference results

The performance of the new (MT-oriented) method to train HMM-based PoS taggers for MT is evaluated on three different source languages —Spanish, French and

Occitan— the target language being Catalan. Here, the results achieved by the following MT "setups" used as reference are reported:

Baum-Welch: The HMM-based PoS tagger is trained by following the classical unsupervised approach (see Section B.3) and then used to disambiguate or to translate (depending on the error measure being reported) a test corpus. Training is done by initializing the parameters by means of Kupiec's method (Kupiec 1992; Manning and Schütze 1999, p. 358; see Section B.3.6) and reestimating the model through the Baum-Welch algorithm (Baum, 1972). When reestimating the HMM parameters the log-likelihood of the training corpus is calculated after each iteration; the iterative reestimation process ends when the difference between the log-likelihood of the last iteration and the previous one is below a certain threshold empirically determined.

Supervised: The HMM-based PoS tagger is trained via the MLE method (see Section B.4) from hand-tagged corpora and then used to disambiguate or to translate a test corpus. Results using this setup are only provided for Spanish, as no hand-tagged corpora are available for the other two languages.

TLM-best: Instead of using an SL PoS tagger, a TL model M_{TL} is used at *translation time* to always select the most likely translation into the TL. To that end, all possible disambiguation paths of each text segment are translated into the TL and scored against the TL model M_{TL}. Note that this MT setup is not feasible for real applications, such as online MT, because the number of disambiguation paths per segment, and consequently the number of translations to perform, grows exponentially with the segment length.

The results achieved by the Baum-Welch MT setup may be considered as the baseline to improve upon; in contrast, the results achieved by the TLM-best setup may be considered as an approximate indication of the best results that the MT-oriented training method could achieve, as this last method transfers information about TL trigrams to an SL first-order HMM (bigrams), possibly involving some loss of accuracy.

Table 2.3 shows, on the one hand, the WERs and BLEU scores achieved when the PoS tagger used by the MT engine is trained by means of the (unsupervised) Baum-Welch algorithm as explained above and, on the other hand, the results achieved when, instead of a PoS tagger, a TL model is used at translation time to select always the most likely translation into TL (TLM-best, see above).

The Baum-Welch results provided in Table 2.3 for both Spanish and French were obtained using a large corpus (large compared with the corpus sizes used with the MT-oriented method) with around 10 million untagged words. The experiments were performed with five different training corpora and the results provided correspond to the mean and the standard deviation of the WER and the BLEU score for the five

Method	Language	WER (%)	BLEU (%)
Baum-Welch	Spanish	8.4 ± 0.1	85.8 ± 0.1
	French	27.1 ± 0.5	57.0 ± 0.6
	Occitan	8.3	84.4
TLM-best	Spanish	6.6	88.3
	French	25.0	60.2
	Occitan	6.7	87.7

Table 2.3: WERs and BLEU scores achieved for the three languages when the PoS tagger used by the MT engine is trained through the (unsupervised) Baum-Welch algorithm (see Section 2.5.2), and when a (Catalan) TL model is used at translation time to score all possible translations and then select the most likely one (TLM-best). The WERs and BLEU scores provided for the Spanish and French (Baum-Welch trained) PoS taggers correspond to the mean and the standard deviation after training with 5 different corpora.

Method	PoS tagging ER (%)	WER (%)	BLEU (%)
Baum-Welch	9.7 ± 0.1	8.4 ± 0.1	85.8 ± 0.1
Supervised	4.9	6.6	88.2

Table 2.4: PoS tagging error rate, WER and BLEU score for the Spanish PoS tagger when it is trained by means of the Baum-Welch (unsupervised) algorithm using untagged corpora, and when it is trained in a supervised way through the MLE method (see Section B.4 on page 125) using a tagged corpus. The error rates reported for the Baum-Welch algorithm correspond to the mean and the standard deviation of the error rates achieved after training with 5 disjoint corpora.

corpora. The Occitan PoS tagger was trained using a small corpus with around 300 000 words as a larger corpus was not available for this less-resourced language.[6]

Table 2.4 shows the PoS tagging error rate, the WER and the BLEU score attained when training the Spanish PoS tagger with the classic unsupervised method, the Baum-Welch algorithm, and with the, also classical, supervised MLE approach. For the latter a Spanish hand-tagged corpus with around 21 500 words was used. This corpus is independent from the corpus used for evaluation. As before, the error rates provided for the Baum-Welch algorithm correspond to the mean and the standard deviation of the error rates achieved after training with five disjoint corpora as explained above.

Although the MT-oriented method is aimed at producing PoS taggers for MT, the PoS tagging error rate is provided so as to show how the PoS tagging error rate correlates with the WER and the BLEU score. Unfortunately, neither tagged corpus was available to train the French and the Occitan PoS taggers in a supervised way, nor to evaluate them.

[6]Note that Occitan has a reduced community of native speakers of the order of one million people.

2.5.3 Use of a complete structural transfer MT system

This section studies the translation performance into Catalan of the MT-oriented training method after training the PoS taggers for Spanish, French and Occitan; moreover, for the Spanish PoS tagger it also studies the PoS tagging performance.

Text segmentation

An adequate strategy for SL text segmentation is necessary as described in Section 2.3. The strategy followed in this experiment consists of segmenting at unambiguous words whose PoS tag is not present in any structural transfer pattern, or at unambiguous words appearing in patterns that cannot be matched in the lexical context in which they appear. To do so, for every pattern involving an unambiguous word, we look at the surrounding words that could be matched in the same pattern, and segmentation is performed only if none of these words have a PoS tag causing the transfer pattern to be matched. For example, to determine if an unambiguous word with the PoS tag "noun" is a segmentation point, all transfer patterns for the corresponding language pair are examined. Suppose that the tag "noun" only appears in these two structural transfer patterns: "article–noun" and "article–noun–adjective". The segmentation will be performed only if the previous and the next word cannot be assigned the "article" and "adjective" PoS tags, respectively.

In addition, an exception is taken into account; no segmentation is performed at words which start a multi-word whose translation could be contracted into a single word (for example in Spanish, *de* followed by *los*, which usually translates as *dels* (=*de*+*els*) into Catalan). Unknown words are also treated as segmentation points, even though they are considered ambiguous, since the *lexical transfer* has no bilingual information for them and no *structural transfer* pattern is activated for them at all.

Results

The experiments were conducted using five disjoint untagged corpora with 500 000 words each for both Spanish and French, and only one corpus with around 300 000 words for Occitan. The use of different training corpora, when available, makes it possible to test if the amount of training corpora needed for convergence is the same for each corpus, and if the MT-oriented method behaves in the same way, in terms of performance, for all of them.

When training, the HMM parameters were estimated, and the resulting performance was recorded, at every 1 000 words in order to see how the method behaved, and to determine the amount of SL text required for convergence.

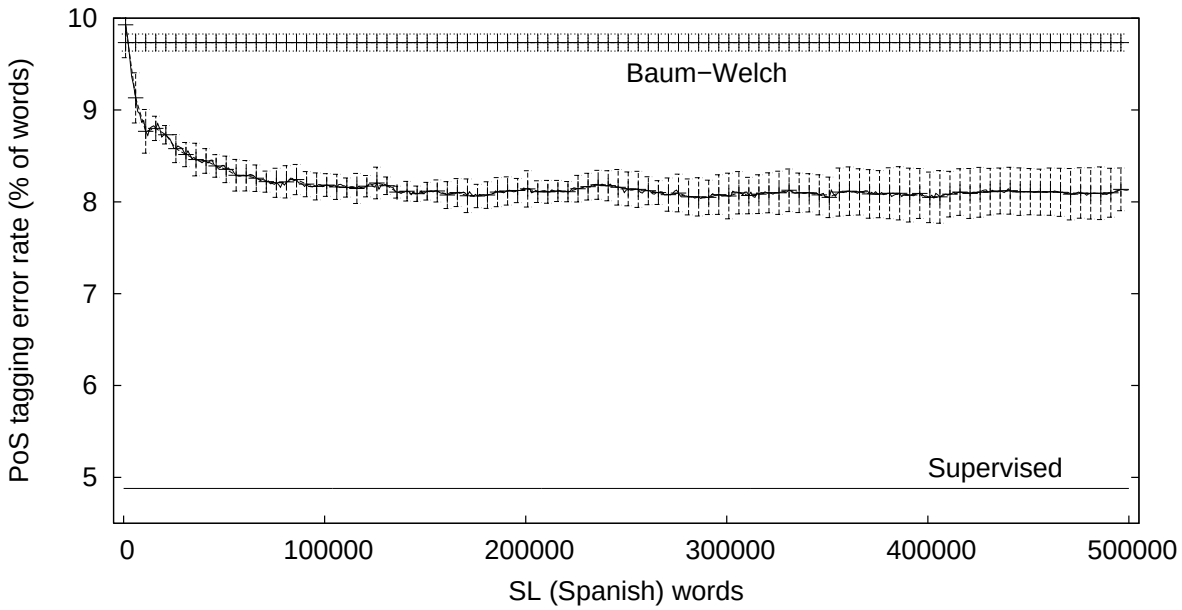


Figure 2.3: Evolution of the mean and standard deviation of the PoS tagging error rate when training the Spanish PoS tagger, Catalan being the target language (TL). The Baum-Welch and supervised results reported in Table 2.4 are displayed for reference (they are independent of the number of SL words).

Figure 2.3 shows, for the five disjoint corpora used to train the Spanish PoS tagger, the evolution of the mean and the standard deviation of the PoS tagging error rate; Figure 2.4 shows the corresponding evolution of the mean and the standard deviation for the WER and the BLEU score. In both figures the Baum-Welch and supervised results reported in Table 2.4, and the TLM-best results (only for the translation quality measures) reported in Table 2.3 are displayed for reference.

As can be seen in Figure 2.3, the performance of the MT-oriented approach, in terms of PoS tagging accuracy, is better than the performance achieved when training via the Baum-Welch algorithm, but it goes about one third of the way toward the tagging performance achieved when training in a supervised way from hand-tagged corpora. Nevertheless, as can be seen in Figure 2.4, the translation quality achieved by the MT-oriented method is almost equal to that achieved by the supervised training method, and very close to that achieved by the TLM-best setup.

The fact that this new (MT-oriented) method achieves a translation quality which is comparable to that achieved by the supervised method, while PoS tagging performance is worse, may be explained by the free-ride phenomenon (very common in the case of related language pairs such as Spanish–Catalan). As PoS tags involved in a free ride produce the same translation, the method cannot distinguish among those tags while training (recall that the language model is based on surface forms) and the resulting

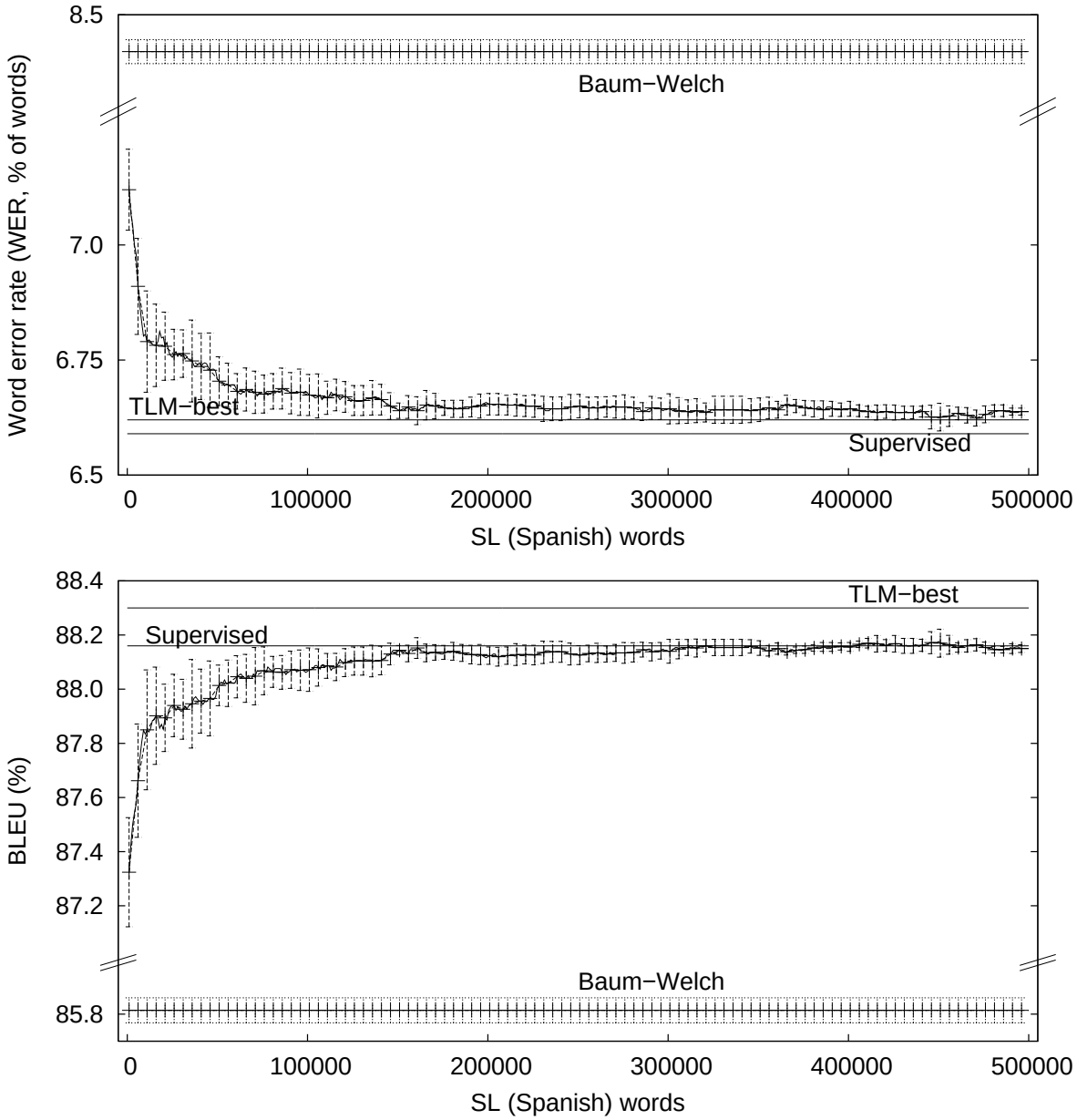


Figure 2.4: Evolution of the mean and standard deviation of the WER (top) and the BLEU score (bottom) when training the Spanish PoS tagger, Catalan being the target language (TL). The Baum-Welch and supervised results reported in Table 2.4, and the TLM-best results reported in Table 2.3 are displayed for reference (they are independent of the number of SL words).

tagger does not correctly tag some words, even if the translation of them is still correct (see Table 2.2).

Figures 2.5 and 2.6 show, respectively, the evolution of the mean and the standard deviation of the WER and the BLEU scores for the 5 disjoint corpora used to train the French PoS tagger, and the evolution of the WER and the BLEU score when training the Occitan PoS tagger, both of them for translation into Catalan. In both cases the Baum-Welch and the TLM-best results reported in Table 2.3 are displayed for reference.

As can be seen in Figures 2.4, 2.5 and 2.6 the MT-oriented training method does not need too much text to converge and the translation performance it achieves is better than that achieved by the Baum-Welch algorithm. Moreover, as Figure 2.5 shows, the translation performance achieved for French by the MT-oriented method is slightly better than that achieved when translating using the TLM-best setup. Recall from Section 2.5.2 that the TLM-best setup provides an approximate indication of the best result that may be achieved by the MT-oriented method; the results reported in that figure suggest that the MT-oriented method may have a certain generalization ability that makes it able to produce slightly better PoS taggers for MT than it may be initially expected. However, in the case of Occitan, the TLM-best setup provides better results (around 1% better).

Confidence intervals

This section reports the translation performance after training with one of the training corpora (randomly chosen) and the confidence interval of that performance calculated using the same test corpus as in the previous experiments (see Table 2.2 on page 28) as explained in Section 2.5.1. The use of confidence intervals allows for a better comparison of the MT setups studied and reinforces the results reported in the previous section in which more than one training corpus was used.

Figure 2.7 shows the WER and the BLEU scores achieved by each MT setup, and by the MT system embedding a Spanish PoS tagger trained via the MT-oriented method when translating Spanish into Catalan. WERs and BLEU scores are provided with their respective 95% (longer intervals) and 85% confidence intervals computed for the corresponding test corpus by repeatedly calculating WERs or BLEU scores (depending on the error measure) from a test corpus randomly drawn with replacement from the original one (see Table 2.2) for 1 000 times. A test corpus built in this way has exactly the same number of sentences than the original one, thus some sentences may appear more than once whereas others may not appear at all.

As can be seen in Figure 2.7 the results achieved by each MT setup provide confidence intervals of similar length. Those intervals show the range of values within the reported measure lies with probability 0.95 or 0.85 (depending on which confidence interval we pay attention to) for test sets of 457 sentences (see Table 2.2).

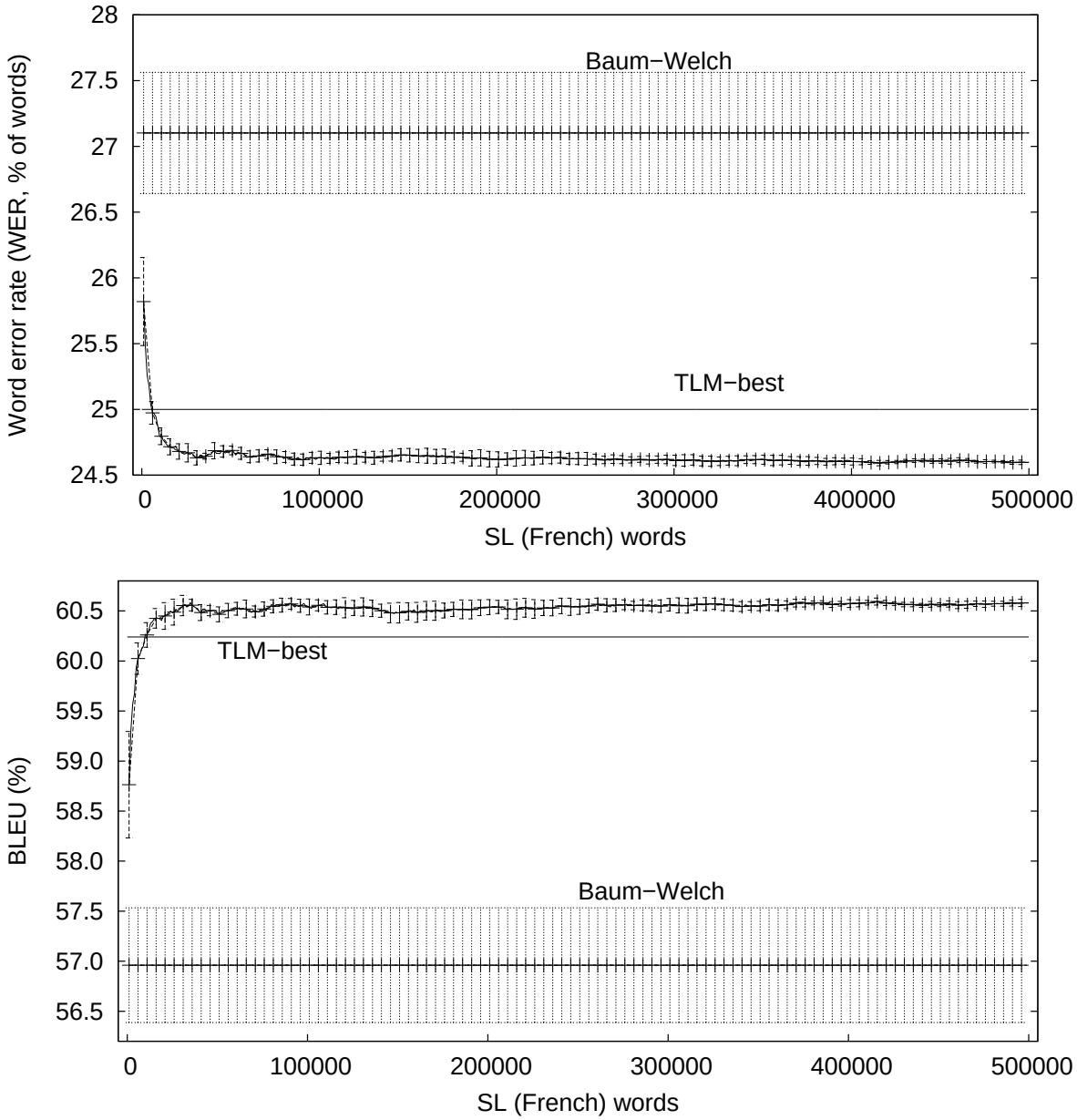


Figure 2.5: Evolution of the mean and standard deviation of the WER (top) and the BLEU score (bottom) when training the French PoS tagger, Catalan being the target language (TL). The Baum-Welch and TLM-best results reported in Table 2.3 are given for reference (they are independent of the number of SL words).

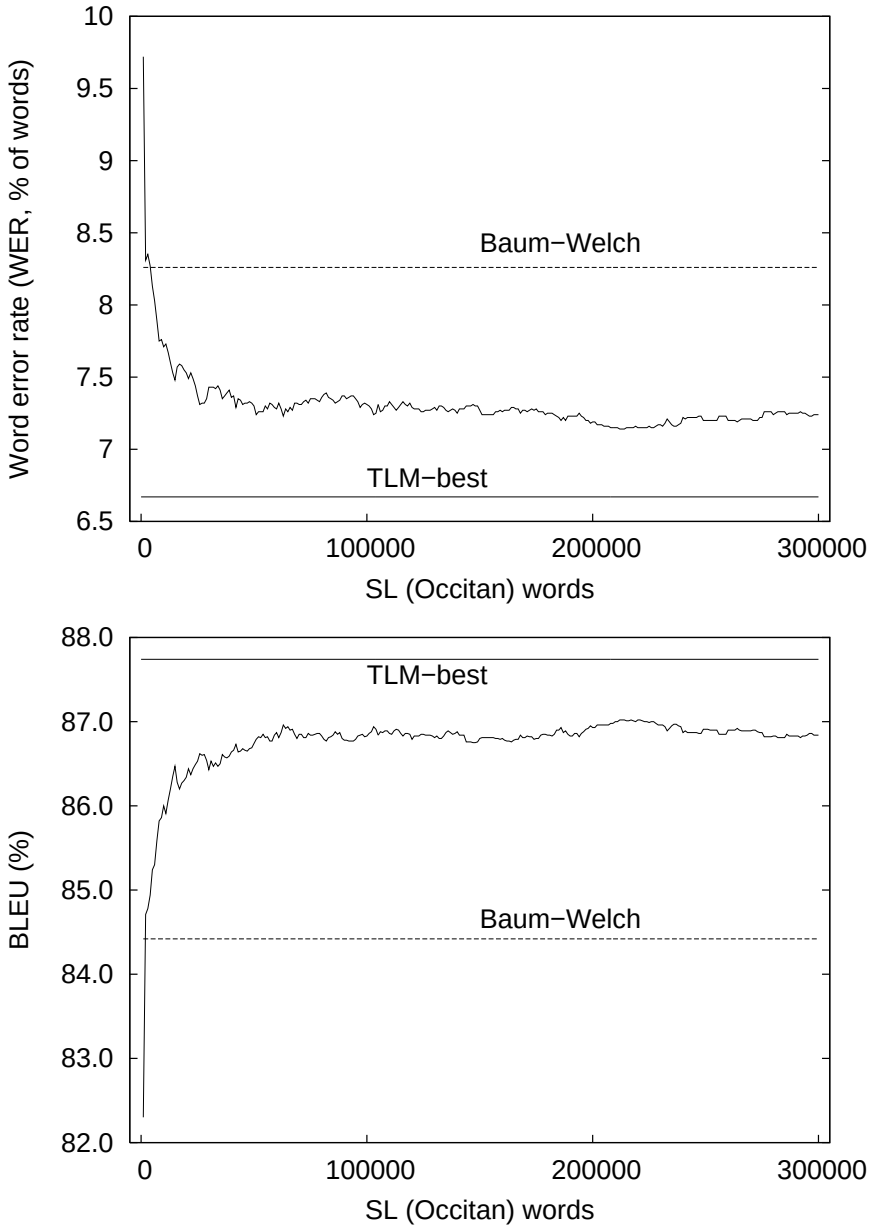


Figure 2.6: Evolution of the WER (top) and the BLEU score (bottom) when training the Occitan PoS tagger, Catalan being the target language (TL). The Baum-Welch and TLM-best results reported in Table 2.3 are given for reference (they are independent of the number of SL words).

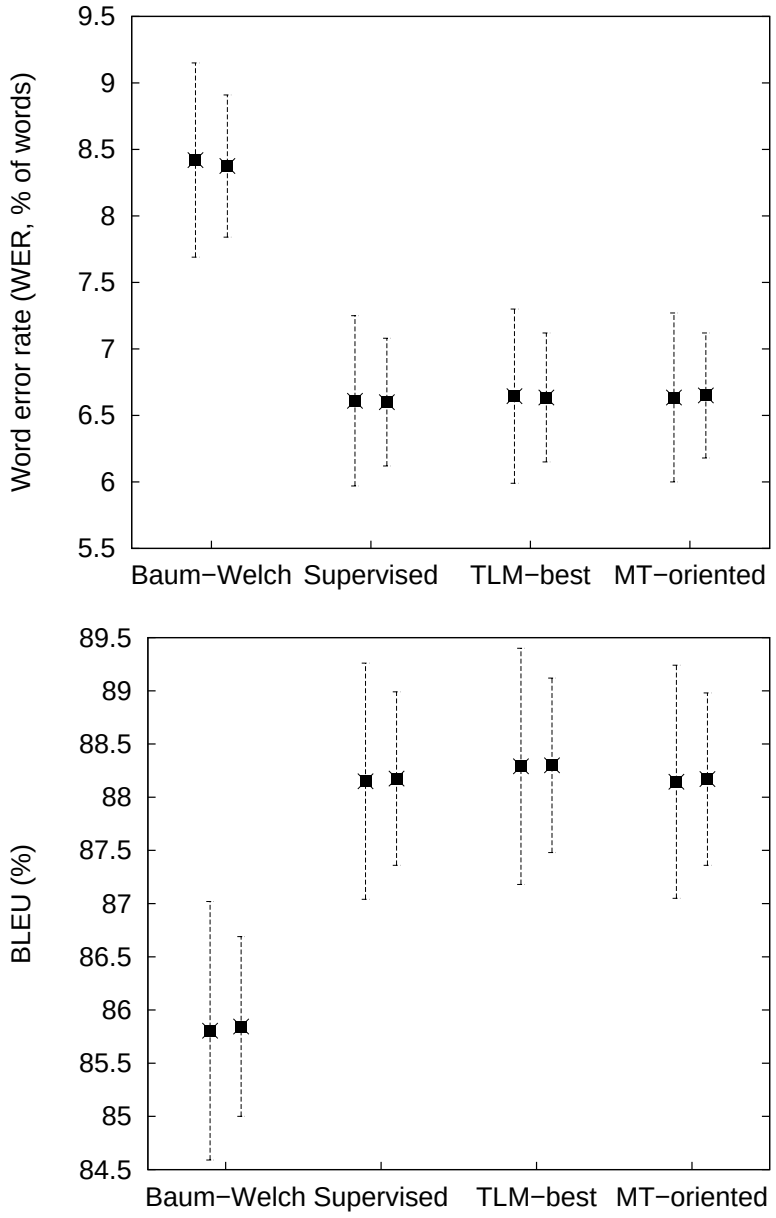


Figure 2.7: WERs and BLEU scores, with their respective 95% (longer intervals) and 85% confidence intervals for test sets of 457 sentences, obtained for the Spanish-to-Catalan translation by each MT setup and by the MT system embedding a Spanish PoS tagger trained via the MT-oriented training method.

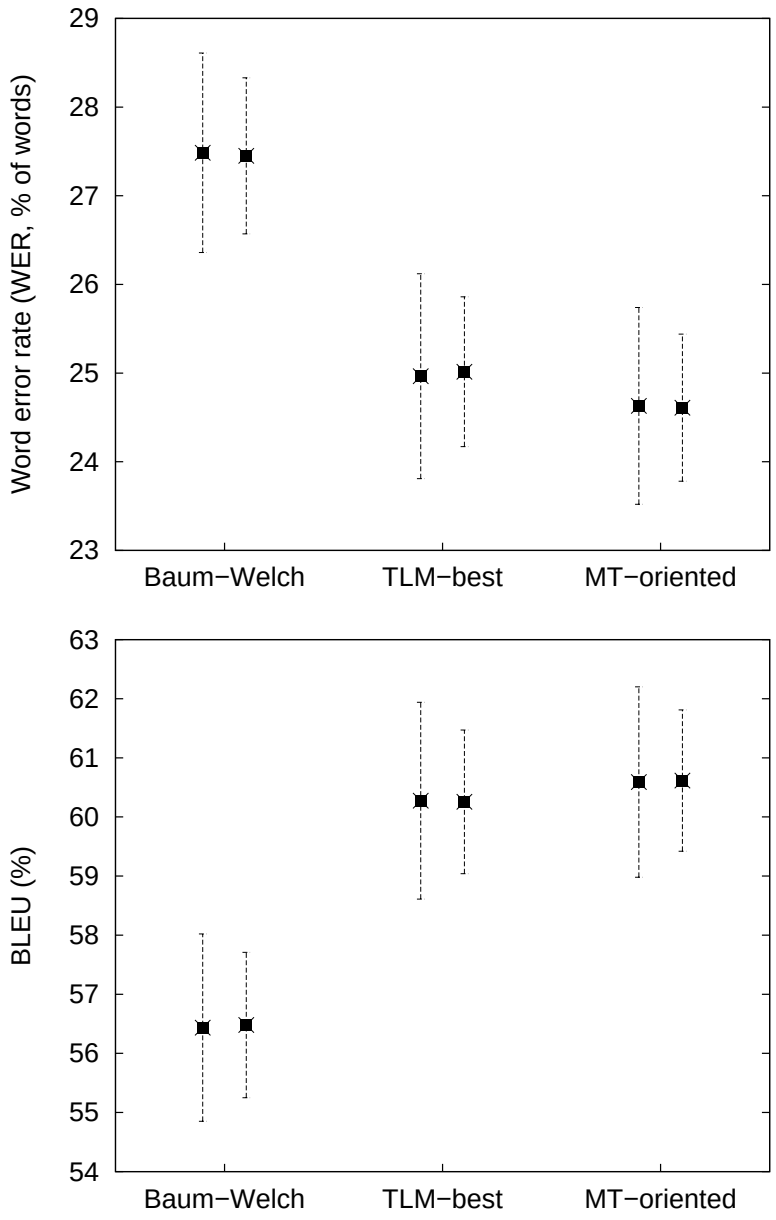


Figure 2.8: WERs and BLEU scores, with their respective 95% (longer intervals) and 85% confidence intervals for 387-sentence test sets, obtained for the French-to-Catalan translation by each MT setup and by the MT system embedding a French PoS tagger trained via the MT-oriented training method.

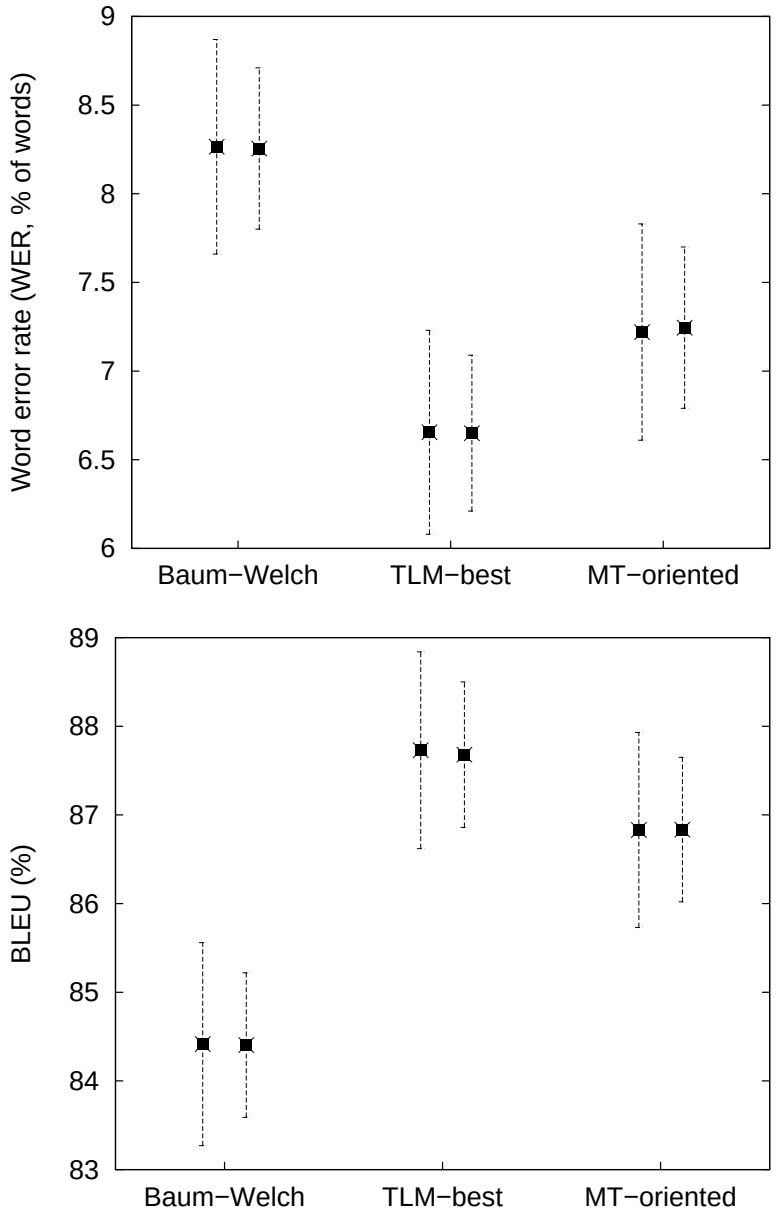


Figure 2.9: WERs and BLEU scores, with their respective 95% (longer intervals) and 85% confidence intervals for evaluation corpora of 538 sentences, obtained for the Occitan-to-Catalan translation by each MT setup and by the MT system embedding an Occitan PoS tagger trained through the MT-oriented training method.

Figures 2.8 and 2.9 show for French-to-Catalan and Occitan-to-Catalan translation, respectively, the WER and BLEU score achieved by each MT setup, and by the MT system embedding an SL PoS tagger trained via the MT-oriented method. As for the Spanish-to-Catalan translation, WERs and BLEU scores are shown with their respective 95% (longer intervals) and 85% confidence intervals computed as described above.

As can be seen in both figures, all of the MT setups provide confidence intervals of similar length as for Spanish-to-Catalan translation (Figure 2.7). Moreover, it should be noted that for the three languages the range of values of the translation performance achieved by the Baum-Welch algorithm does not overlap with that of the MT-oriented training method, except for the 95% confidence interval provided for the WER for the Occitan-to-Catalan translation.

2.5.4 Use of a null structural transfer MT system

In the experiments reported in the previous section a full structural transfer MT system was used. Because of this, information about transfer patterns had to be taken into account when segmenting in order to make each segment independent of adjacent ones. This section presents a set of experiments conducted when reducing the structural transfer of the corresponding language pair to a minimum (context-free word-for-word) null structural transfer component. That is, in this section, results are reported when training runs using an MT system in which the structural transfer module has no transfer patterns and, consequently, processes the input word for word without taking context into account. This experiment will show that, when the languages involved are closely related, using a null structural transfer module does not seriously affect the resulting performance of the MT-oriented training method. Note that a null structural transfer component is used while training, but full structural transfer is used for the evaluation of translation performance of the MT system when embedding a given PoS tagger.

Text segmentation

As transfer patterns have been removed from the structural transfer module, each word is treated independently of the adjacent ones after PoS tagging. This makes it possible for the method to just segment at every unambiguous word, which causes segments to be much smaller and reduces the number of translations to perform per segment. As in the rest of experiments, unknown words are also treated as segmentation points in spite of being ambiguous because no bilingual information is available for them, and therefore unknown words are never translated.

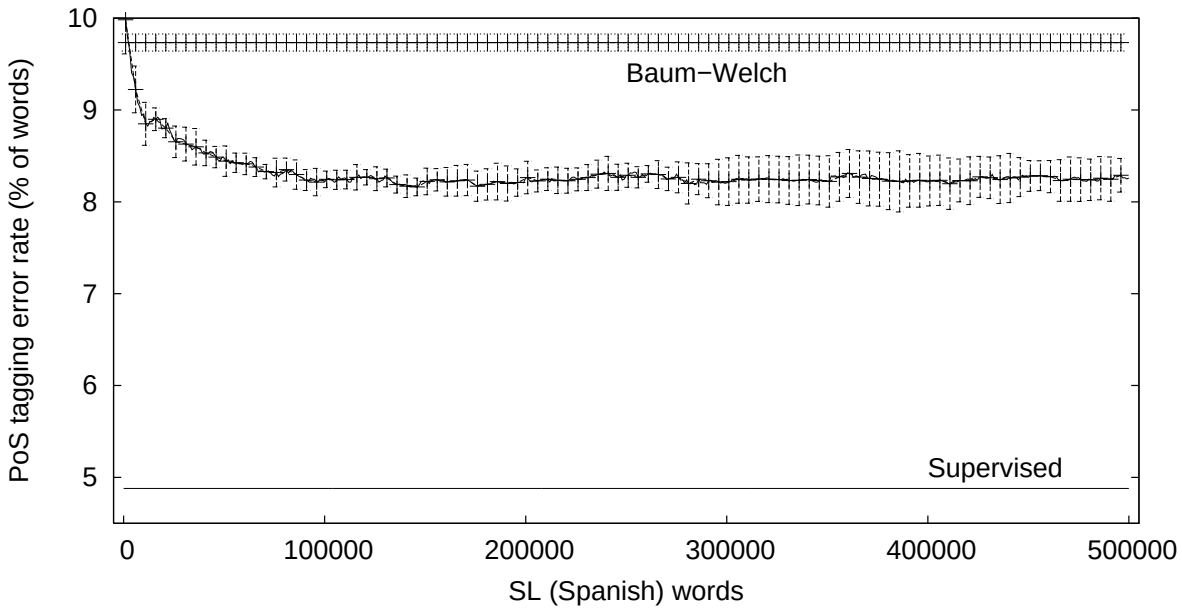


Figure 2.10: Mean and standard deviation for the PoS tagging error when a null structural transfer is used while training the Spanish PoS tagger, Catalan being the target language (TL). Baum-Welch and supervised results are given for reference. Compare with Figure 2.3 in which the same corpora are considered but using a full structural transfer in the training phase; both figures are very similar.

Results

Figures 2.10 and 2.11 show, for the 5 disjoint corpora used to train the Spanish PoS tagger through a null structural transfer MT system, the evolution of the mean and the standard deviation of the PoS tagging error rate, and the evolution of the mean and the standard deviation of the WER and the BLEU score achieved by the MT system embedding the resulting PoS tagger, respectively. Recall, however, that the full structural transfer MT system is still used when evaluating. As in the experiments reported in the previous section, the Baum-Welch, supervised, and TLM-best results are displayed for reference. Note, however, that, for direct comparison with the MT-oriented method, in this case the TLM-best result was calculated by using a null structural transfer for selecting the disambiguation path that produces the most likely translation, but using the full one to perform the translation finally evaluated.

Comparing Figures 2.10 and 2.11 with Figures 2.3 and 2.4, in which a full structural transfer MT system is used by the training algorithm, we can see that the results obtained are quite similar even though in this last experiment no actions are performed in order to solve the grammatical divergences between the SL and the TL; this is because Spanish and Catalan are two very related languages. In addition, in this case the obtained PoS tagger performs better than the TLM-best setup.

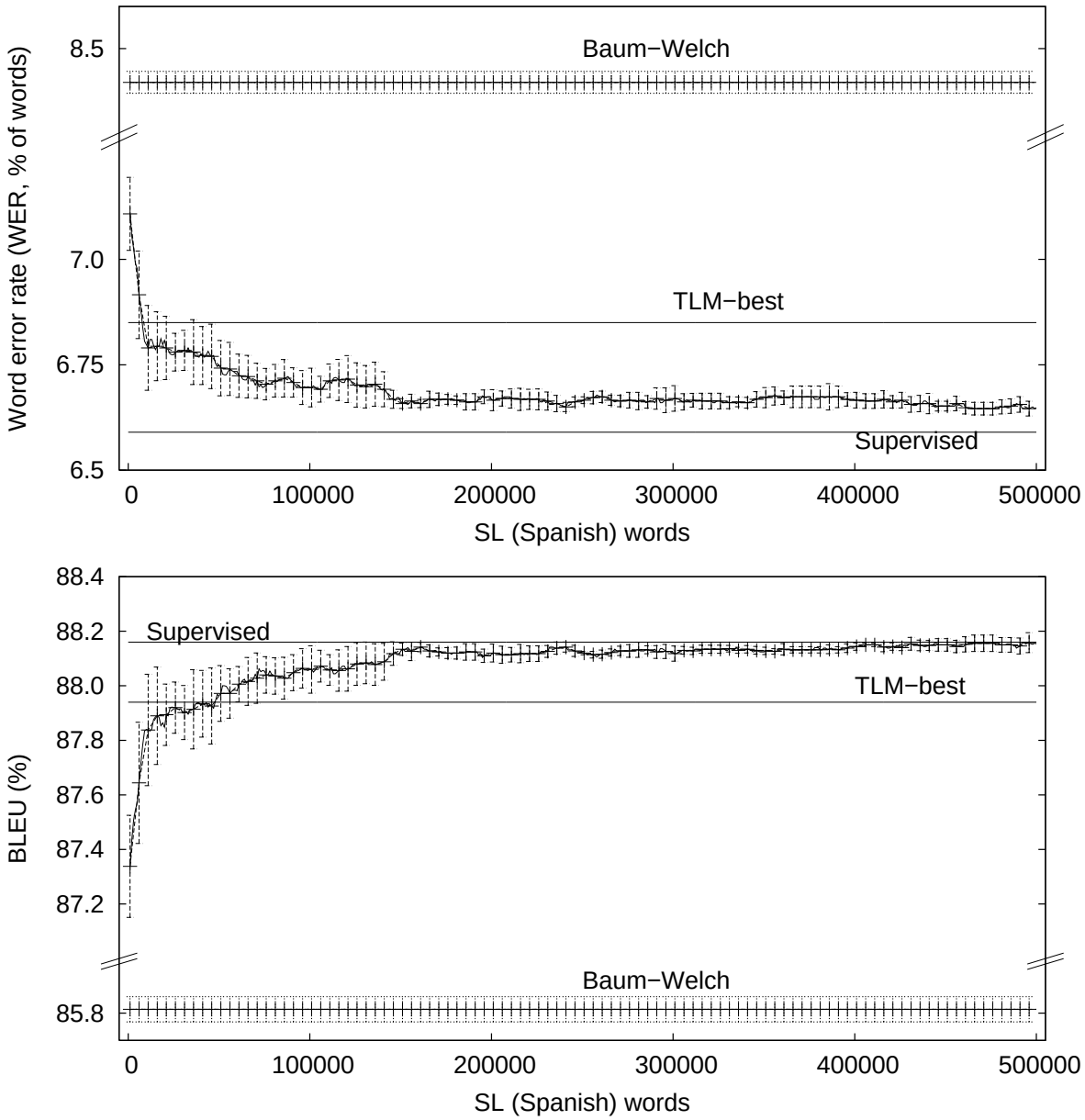


Figure 2.11: Mean and standard deviation of the WER (top) and the BLEU score (bottom) when a null structural transfer is used while training the Spanish PoS tagger, Catalan being the target language (TL). Baum-Welch, supervised and TLM-best results are given for reference. The TLM-best result reported was calculated by using a null structural transfer component when selecting the disambiguation path that produces the more likely translation, but full structural transfer when deriving the evaluated translation. Compare with Figure 2.4 in which the same corpora are considered but using a full structural transfer in the training phase; both figures are very similar.

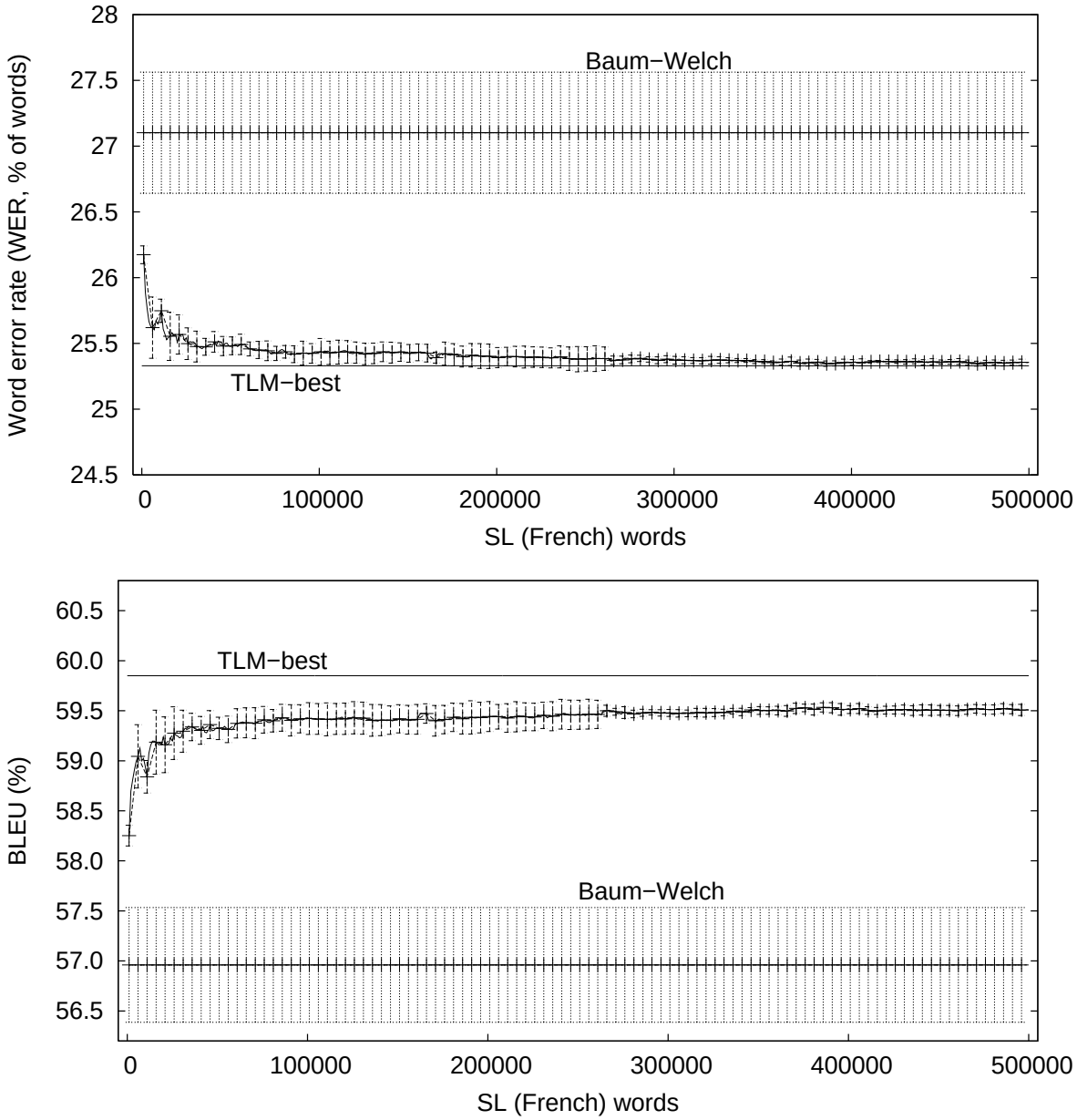


Figure 2.12: Mean and standard deviation of the WER (top) and the BLEU score (bottom) when a null structural transfer is used to train the French PoS tagger, Catalan being the target language (TL). Baum-Welch and TLM-best results are given for reference; as in the rest of figures of this section the TLM-best result reported was calculated by using a null structural transfer component when selecting the disambiguation path producing the most likely translation, but the full one to derive the evaluated translation. Compare with Figure 2.5 in which the same corpora are considered but using a full structural transfer in the training phase.

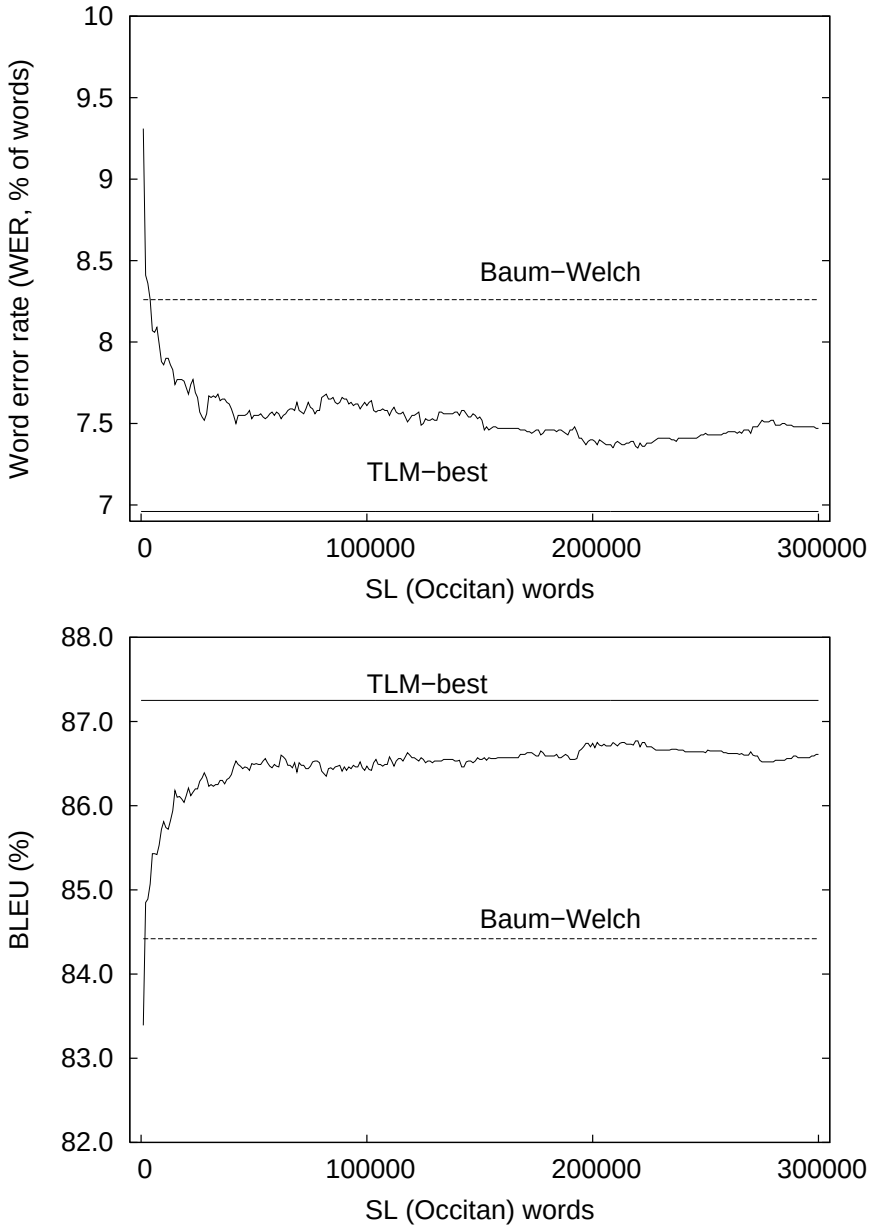


Figure 2.13: WER (top) and BLEU score (bottom) when a null structural transfer component is used to train the Occitan PoS tagger, Catalan being the target language (TL). The WER and the BLEU score after training the Occitan PoS tagger via the Baum-Welch algorithm on the same corpus, and the TLM-best result (calculated with a null structural transfer) are given for reference. Compare with Figure 2.6 in which the same corpus is considered but using a full structural transfer in the training phase.

The experiments conducted with the Occitan–Catalan and French–Catalan language pairs (see Figures 2.12 and 2.13, respectively) show that the PoS taggers trained by using a null structural transfer MT system are worse than those obtained by using the full transfer MT system. For French the WER is around 0.8% worse, whereas for Occitan the WER is only around 0.2% worse; BLEU scores show the same behaviour. Note that, although these results are slightly worse than those reported when a full structural transfer MT system is used for training, the resulting PoS taggers are still better than those trained using the Baum-Welch algorithm.

The fact that the results achieved when using a null structural transfer component are different for different language pairs, compared with the results achieved when using full structural transfer, gives an indication of how related two languages are. Note that when no transfer rules are taken into account, no actions are performed to treat the grammatical divergences between the languages involved. These experiments suggest that Spanish and Catalan are more related than Occitan and Catalan, or French and Catalan.

Confidence intervals

With respect to the confidence intervals, Figures 2.14, 2.15 and 2.16 show the WER and BLEU scores, with their respective 95% and 85% confidence intervals, achieved when translating Spanish, French and Occitan, respectively, into Catalan when the PoS taggers being used to carry out the translation are trained through a null structural transfer MT system. All figures give confidence intervals with similar width for the three language pairs, and similar to those given when training using a full structural transfer MT system. Note that, as the Spanish–Catalan language pair shows similar results for both experiments, there is no intersection of the confidence intervals for the Baum-Welch algorithm and the MT-oriented training method.

In the case of French and Occitan (Figures 2.15 and 2.16), the 95% confidence intervals for the Baum-Welch and the MT-oriented method overlap, both for the WER and the BLEU score, because the results when a null structural transfer component is used in the training phase are worse for these two languages; note that the overlapping is more significant in the case of Occitan. However, the 85% confidence intervals for BLEU do not overlap. That is, with probability 0.85 the performance of the MT-oriented training method is still better (according to BLEU) than that of the Baum-Welch algorithm for test sets of the size of those used in the evaluation (see the number of sentences of the evaluation corpora on Table 2.2, page 28).

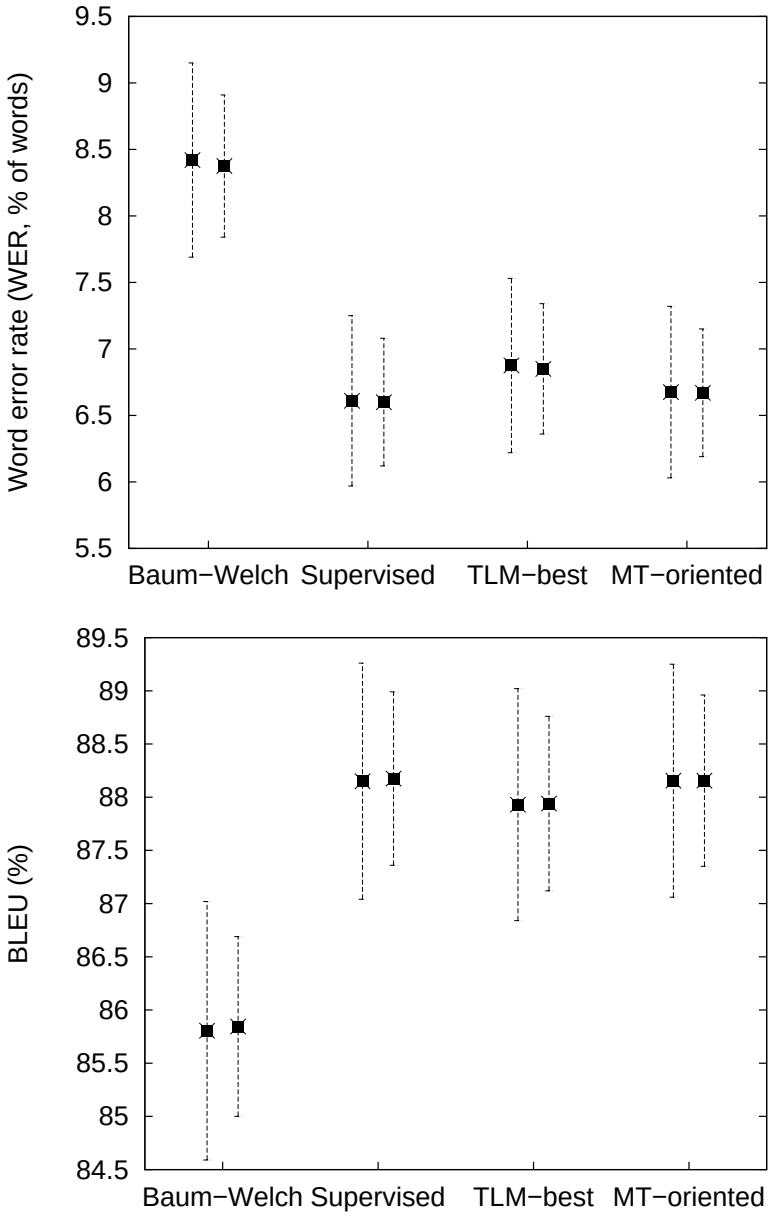


Figure 2.14: WERs and BLEU scores, with their respective 95% (longer intervals) and 85% confidence intervals, achieved for Spanish-to-Catalan translation by each MT setup and by the MT system embedding a Spanish PoS tagger trained via the MT-oriented training method when a null structural transfer component is used while training.

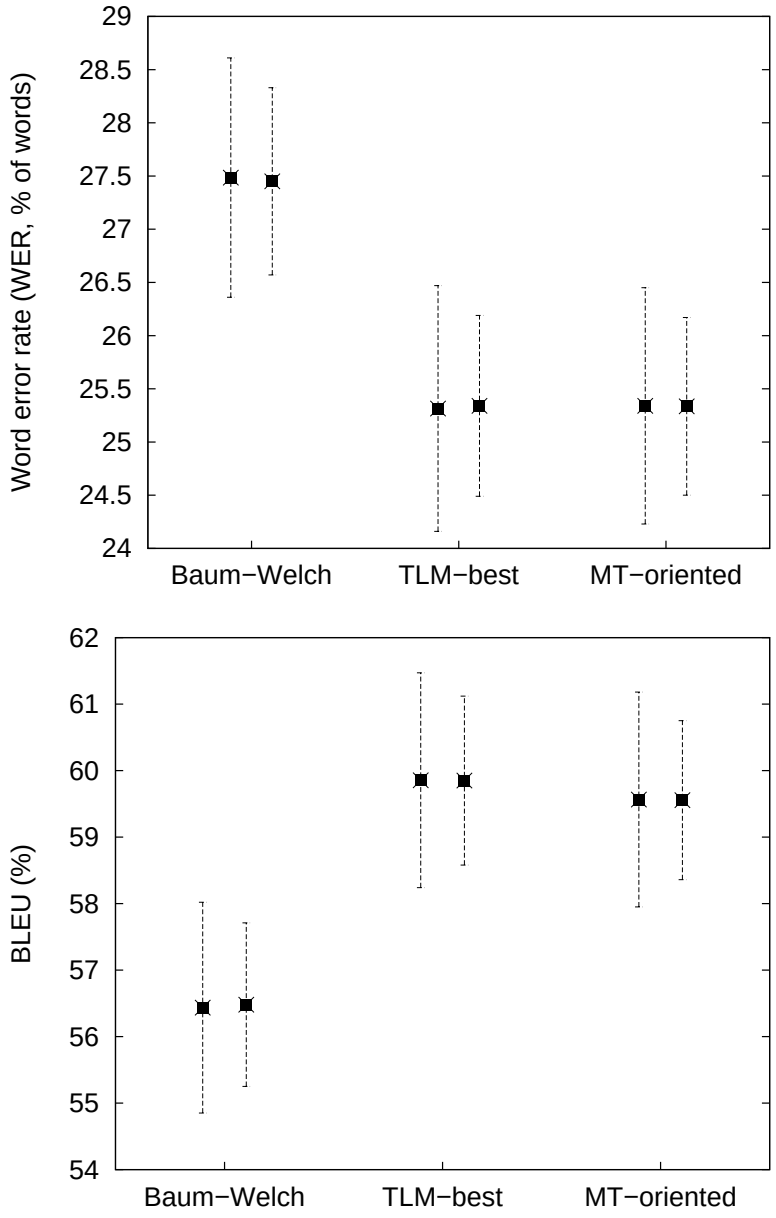


Figure 2.15: WERs and BLEU scores, with their respective 95% (longer intervals) and 85% confidence intervals, achieved for French-to-Catalan translation by each MT setup and by the MT system embedding a French PoS tagger trained via the MT-oriented training method when a null structural transfer component is used in the training phase.

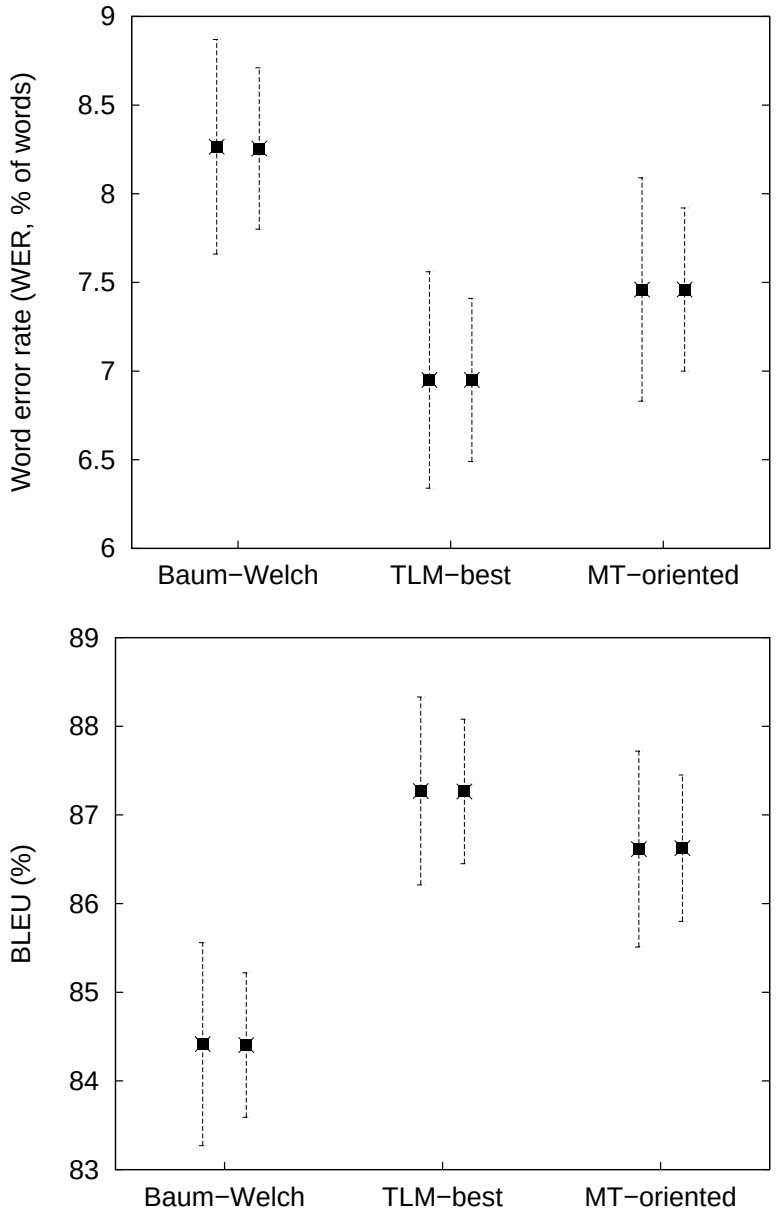


Figure 2.16: WERs and BLEU scores, with their respective 95% (wider intervals) and 85% confidence intervals, achieved for Occitan-to-Catalan translation by each MT setup and by the MT system embedding a PoS tagger trained via the MT-oriented training method when a null structural transfer component is used for training.

2.6 Discussion

This chapter has explored the use of TL information to train HMM-based PoS taggers to be used in MT. In the experiments, this new MT-oriented approach has been tested on three different languages (Spanish, French and Occitan), all being translated into Catalan. The performance of the MT-oriented approach has been compared with three different MT "setups" or configurations: the use of a PoS tagger trained through the standard unsupervised approach (Baum-Welch), the use of PoS tagger trained in a supervised way from hand-tagged corpora (only for Spanish), and the use of a TL model at translation time (instead of a PoS tagger) to select always the most likely translation into TL (TLM-best). The Baum-Welch MT setup can be considered as the baseline whose results are improved upon, since the MT-oriented method also runs in an unsupervised manner, while the TLM-best may be seen as an indication of the best results that this new training method may achieve (see Section 2.5.2).

For all the three languages the MT-oriented method gives better results than the Baum-Welch trained PoS tagger, and results of the order of those achieved by the TLM-best setup. Note that, although the MT-oriented training algorithm also uses a TL model to score translations, this is only done for training, never at translation time; therefore, the PoS tagger is as fast as any other HMM-based PoS tagger. However, the use of the TLM-best setup makes translation much slower, since all possible disambiguations of a given text segment must be translated and scored against a TL model before selecting the most likely translation, this makes the TLM-best setup not feasible for real applications such as online MT.

The results on the Spanish language show that the translation quality achieved by the MT system embedding a PoS tagger trained via this new unsupervised method is comparable to that achieved by the same MT system when embedding a PoS tagger trained in a supervised manner from hand-tagged corpora. However, as far as the PoS tagging accuracy is concerned, the MT-oriented method performs better than the classical unsupervised approach (around 1.6 percentage points) but worse than the supervised one (3.2 percentage points worse). This may be due to the existence of free rides, as the MT-oriented method cannot distinguish between PoS tags leading to the same translation. Therefore, it can be concluded that, as expected, the MT-oriented method is a good choice to train PoS taggers for MT, but not as good as the supervised one to train general-purpose PoS taggers to be used in other natural language processing applications.

In the experiments two different MT systems have been used in the training phase, one having a structural transfer module that performs some operations, such as gender and number agreement or word reordering to meet the TL grammatical rules (see Section 2.5.3), and another that does not perform any structural transfer operation (see Section 2.5.4). The latter can be said to process each word independently from the adjacent ones after PoS tagging. It has been shown that the results achieved in both cases are quite similar for Spanish, and less similar for Occitan and French; it can

be concluded that Spanish and Catalan are more related than the other two language pairs.

Finally, it is worth mentioning that the main disadvantage of the method presented in this chapter is that the number of translations to perform for each SL text segment grows exponentially with segment length. The following chapter presents a very simple technique that can be used to overcome this problem without affecting the final PoS tagging and translation performance.

Chapter 3

Pruning of disambiguation paths

The main disadvantage of the MT-oriented training method introduced in the previous chapter is that the number of translations to perform by the training algorithm grows exponentially with segment length, translation being the most time-consuming task. This chapter presents a method that uses *a priori* knowledge obtained in an unsupervised manner to prune unlikely disambiguations in each text segment, so that the number of translations to be performed during training is reduced. The experimental results show that this pruning method drastically reduces the amount of translations performed during training without degrading the translation quality achieved by the RBMT system embedding the resulting PoS tagger.

3.1 Pruning method

The objective of the method introduced in this section is to reduce as much as possible the number of translations to perform per segment during training without degrading the translation performance achieved by the RBMT system embedding the resulting PoS tagger. The disambiguation pruning method is based on *a priori* knowledge, that is, on an initial model $\hat{M}_{\text{tag}}^{[0]}$ of SL tags. The assumption here is that any reasonable model of SL tags may be helpful to choose a subset $T'(s)$ of the set of all possible disambiguation paths $T(s)$ of each segment s, such that the winner is in that subset. Therefore, there is no need to translate all possible disambiguation paths of each segment into the TL, but only the most *promising* ones.

The initial model $\hat{M}_{\text{tag}}^{[0]}$ of SL tags to be used can be either an HMM or any other model whose parameters are obtained by means of a statistically sound method. Nevertheless, using an HMM as an initial model allows the method to dynamically evolve, obtaining a new model $\hat{M}_{\text{tag}}$ that is the result of integrating new evidence collected during training (see Section 3.2 for more details).

The pruning of disambiguation paths for a given SL text segment s is carried out as follows: first, the *a priori* likelihood $\hat{P}_{\text{tag}}(\boldsymbol{g}|s)$ of each possible disambiguation path $\boldsymbol{g}$ of segment s in the tagging model $\hat{M}_{\text{tag}}$ is calculated (see below); then, the subset of disambiguation paths to take into account is determined according to the calculated *a priori* likelihoods.

Let $U(s)$ be an ordered set of all possible disambiguation paths of the SL segment s; disambiguation paths $\boldsymbol{g} \in U(s)$ are ordered in decreasing order of their *a priori* likelihood, that is, $U(s) = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{|T(s)|}\}$ with $\boldsymbol{g}_i \in T(s) : \ 1 \leq i \leq |T(s)|$, and $\hat{P}_{\text{tag}}(\boldsymbol{g}_i|s) \geq \hat{P}_{\text{tag}}(\boldsymbol{g}_{i+1}|s)$.

To decide which disambiguation paths to take into account, the pruning algorithm is provided with a mass probability threshold $\rho \in [0, 1]$; the subset of disambiguation paths to take into account, $U'(s) = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k\}$ with $k \leq |T(s)|$, must satisfy the following expression:

$$\rho \leq \sum_{i=1}^{k} \hat{P}_{\text{tag}}(\boldsymbol{g}|s) \tag{3.1}$$

for the minimum possible value of k. Therefore, after pruning, the MT-oriented training method described in Chapter 2 takes into account the minimum subset of disambiguation paths $\boldsymbol{g} \in T(s)$ needed to reach the mass probability threshold ρ. Note that disambiguation paths $\boldsymbol{g}$ being ruled out will be assumed to have a null probability $P_{\text{tag}}(\boldsymbol{g}|s)$ when estimating the frequency counts $\tilde{n}(\cdot)$ through Equations (2.13), (2.14) and (2.15), that are then used to estimate the HMM parameters as before.

Estimation of the a priori likelihood

The estimation of the *a priori* likelihood $\hat{P}_{\text{tag}}(\boldsymbol{g}|s)$ of each disambiguation path $\boldsymbol{g}$ is made by considering not only the corresponding segment s itself, but also the context in which segment s appears. Context needs to be taken into account as a consequence of the segmentation strategy because, on the one hand, segments may be started by words that would never appear at the beginning of a well-formed sentence, which makes the vector π with the probability of each path being the initial one completely useless, and, on the other hand, because segments may be too short to get an accurate *a priori* estimate of the likelihood.

Context is taken into account by calculating the forward and backward probabilities, as in the Baum-Welch EM algorithm (see Equations B.10 and B.12, respectively, in appendix B). After that, the *a priori* likelihood of disambiguation path $\boldsymbol{g} = (\gamma_1 \ldots \gamma_N)$ given segment $s = (\sigma_1 \ldots \sigma_N)$ is calculated through the following equation:

$$\hat{P}_{\text{tag}}(\boldsymbol{g}|s) = \sum_{\gamma_j \in \Gamma} \alpha_-(\gamma_j) a_{\gamma_j \gamma_1} b_{\gamma_1}(\sigma_1) \prod_{i=2}^{N} a_{\gamma_{i-1} \gamma_i} b_{\gamma_i}(\sigma_i) \sum_{\gamma_j \in \Gamma} a_{\gamma_N \gamma_j} b_{\gamma_j}(\sigma_j) \beta_+(\gamma_j), \tag{3.2}$$

where $\alpha_{-}(\gamma_j)$ refers to the forward probability of PoS tag γ_j for the word preceding the first one in the segment being considered, and $\beta_{+}(\gamma_j)$ refers to the backward probability of PoS tag γ_j for the first word after the last one of segment s. For efficiency, the computation of $\alpha_{-}(\gamma_j)\,\forall\gamma_j \in \Gamma$ starts at the first unambiguous word preceding the SL segment s; analogously, the calculation of $\beta_{+}(\gamma_j)\,\forall\gamma_j \in \Gamma$ ends at the first unambiguous word after segment s.

3.2 Updating the model

This section explains how the model $\hat{M}_{\text{tag}}$ used for pruning can be updated during training so that it integrates new evidence collected from the TL. The idea is to periodically estimate during training an HMM using the counts collected from the TL (as explained in Chapter 2, Section 2.2), and to mix the resulting HMM with the initial one, $\hat{M}^{[0]}_{\text{tag}}$; the mixed HMM becomes the new model $\hat{M}_{\text{tag}}$ used for pruning.

The initial model and the model obtained during training are mixed so that the estimate of *a priori* likelihoods is the best possible at each moment; mixing affects both transition and emission probabilities.

Let $\boldsymbol{\theta} = (a_{\gamma_1\gamma_1}, ..., a_{\gamma_{|\Gamma|}\gamma_{|\Gamma|}}, b_{\gamma_1}(\sigma_1), ..., b_{\gamma_{|\Gamma|}}(\sigma_{|\Sigma|}))$ be a vector containing all the parameters of a given HMM. The parameters of the initial HMM and those of the new one can be mixed through the following linear combination:

$$\boldsymbol{\theta}(x) = \varphi(x)\,\boldsymbol{\theta}^{\text{TL}}(x) + (1 - \varphi(x))\,\boldsymbol{\theta}^{[0]}, \tag{3.3}$$

where $\boldsymbol{\theta}(x)$ refers to the HMM parameters after mixing the two models when a fraction x of the training corpus has been processed; $\boldsymbol{\theta}^{\text{TL}}(x)$ refers to the HMM parameters estimated by means of the MT-oriented method described in Chapter 2 after processing a fraction x of the SL training corpus; and $\boldsymbol{\theta}^{[0]}$ refers to the parameters of the initial HMM ($\hat{M}^{[0]}_{\text{tag}}$). Function $\varphi(x)$ assigns a weight to the model estimated using the counts collected from the TL ($\boldsymbol{\theta}^{\text{TL}}$). This monotonically increasing weight function is made to depend on the fraction x of the SL corpus processed so far so that $\varphi(0) = 0$ and $\varphi(1) = 1$.

3.3 Experiments

In the experiments reported in Chapter 2, all disambiguation paths of each segment were translated into the TL by using the remaining modules of the Apertium MT system that follow the PoS tagger. This section presents a set of experiments conducted to test the approach presented in previous sections of this chapter in order to reduce the amount of translations to be performed per segment and, therefore, the time needed by the training algorithm, without degrading the translation accuracy achieved.

3.3.1 Task

As in Chapter 2 this section focuses on the same three languages —Spanish, French and Occitan— being translated into Catalan by means of the open-source shallow transfer MT platform Apertium (see appendix A). The experiments conducted consist of training an HMM-based PoS tagger for each language using a complete structural transfer MT system in the training phase (as in Section 2.5.3) in conjunction with the pruning method introduced in this chapter. The same linguistic data, tagset, training corpora, and evaluation corpora are used (see Section 2.5.1 for more details).

In order to determine the appropriate mass probability threshold ρ that speeds up the MT-oriented training method without degrading its performance, a set of values for ρ between 0.1 and 1.0 at increments of 0.1 was considered. Note that when $\rho = 1.0$, no pruning is done at all and, therefore, all possible disambiguation paths for each segment are translated into the TL, as in Chapter 2.

Model used for pruning

The initial model used for pruning was computed by means of Kupiec's method (Kupiec 1992, see Section B.3.6), the same unsupervised initialization method used when training via the Baum-Welch EM algorithm. After that, the model is updated during training after every 1,000 words processed as explained in Section 3.2. To this end, the weight function $\varphi(x)$ used in Equation (3.3) was chosen to grow linearly from 0 to 1 with the fraction x of SL corpus processed so far:

$$\varphi(x) = x. \tag{3.4}$$

3.3.2 Results

Figure 3.1 shows, for the different values of ρ used, the mean and standard deviation of the WER (top) and the BLEU score (bottom) achieved by the MT system when embedding the Spanish PoS tagger trained using the 5 disjoint corpora already used in Section 2.5 of Chapter 2. As can be seen, the best results are achieved for values of ρ of 0.8 and 0.9, being indeed slightly better than those achieved when no pruning is performed ($\rho = 1.0$); however, the standard deviation is smaller when no pruning is done, which makes this small improvement irrelevant, but not the benefits of the pruning method.

Figure 3.2 shows the mean and standard deviation of the WER and the BLEU score achieved by the MT system embedding the French PoS tagger being evaluated for all tested values of ρ. Analogously, Figure 3.3 reports the WER and BLEU score of the MT system embedding the Occitan PoS tagger for the same values of ρ. Both languages show a similar behaviour, in accordance with the Spanish language; even

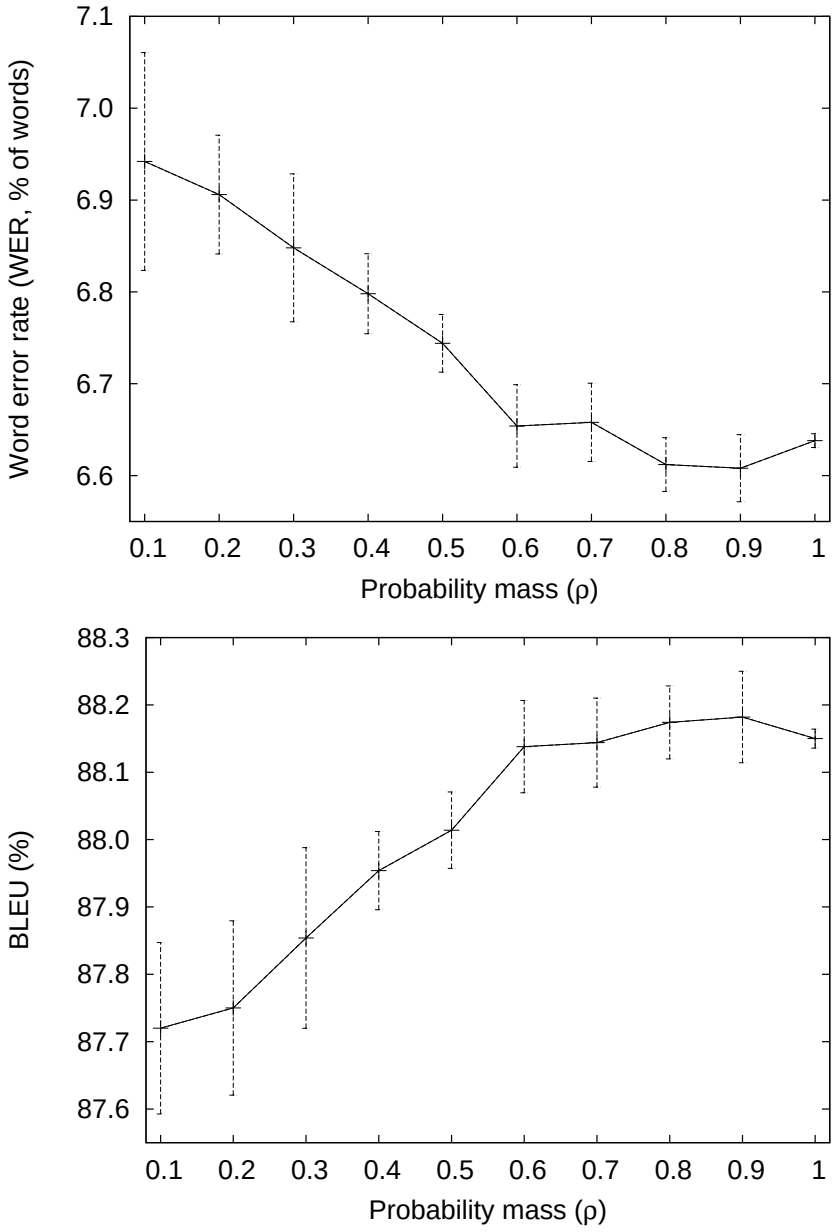


Figure 3.1: For the different values of ρ used, mean and standard deviation of the WER (top) and the BLEU score (bottom) achieved after training the Spanish PoS tagger, Catalan being the TL.

though in the case of French the standard deviation for values of ρ greater than 0.5 is small compared to that of Spanish, and of the order of the standard deviation of the results achieved when no pruning is performed at all ($\rho = 1.0$).

Concerning the translation performance achieved by the Occitan PoS tagger, when $\rho = 0.8$ there is an improvement of around 0.4 percentage points in the WER, i.e. a reduction, and around 0.5 percentage points in the case of BLEU. This improvement, which is more significant than the one achieved for Spanish, may be explained by the fact that fractional counts associated to discarded disambiguation paths are assumed to be null; however, when no pruning is performed these fractional counts may be small, but never null as a consequence of the smoothing applied to the probabilities of the TL model used (see Section 2.4). Nevertheless, it must be noted that the French result is not improved; this may indicate that while training the French PoS tagger some "good" disambiguation paths are ruled out as a consequence of the pruning method.

As in Chapter 2, the PoS tagging performance of the Spanish PoS tagger was evaluated in order to see how it correlates with translation performance, as well as how the pruning method affects the PoS tagging performance. Figure 3.4 shows the mean and the standard deviation of the PoS tagging error rate achieved by the Spanish PoS tagger after training with the 5 disjoint corpora. The behaviour of the PoS tagging error rate is quite erratic as the large standard deviations indicate. This behaviour contrasts with that of the translation performance, and may be explained by the fact that the PoS tagger being learned is specially suited to MT, not for PoS tagging, as the goal is to obtain good translations in the TL, independently of the actual PoS tagging accuracy.

With respect to the evolution of the WER and the BLEU score of the MT system embedding the PoS tagger being trained, Figures 3.5 and 3.6 show the mean and the standard deviation of the WER and the BLEU score, respectively, achieved by the Spanish PoS tagger for two of the different values of the probability mass threshold ρ tested; the two values for ρ shown are the smallest threshold used (0.1), and the threshold that causes the MT system embedding the resulting Spanish PoS tagger to achieve the best translation quality (0.9, see also Figure 3.1). Note that the other two languages show a similar behaviour. As can be seen the evolution of the translation performance is similar to that of using no pruning technique at all (see Figure 2.4 in page 34); therefore, no larger corpora is needed by the MT-oriented training method when pruning unlikely disambiguation paths, since in both cases the amount of training corpora needed for convergence is similar.

As for the model used for pruning, Figure 3.7 shows the evolution of the mean and the standard deviation of the WER (top) and the BLEU score (bottom) of the MT system when embedding the model used for pruning when training the Spanish PoS tagger with $\rho = 0.9$. Recall that this model is a linear combination of the initial model (trained via Kupiec's method) and the model being learned via the MT-oriented

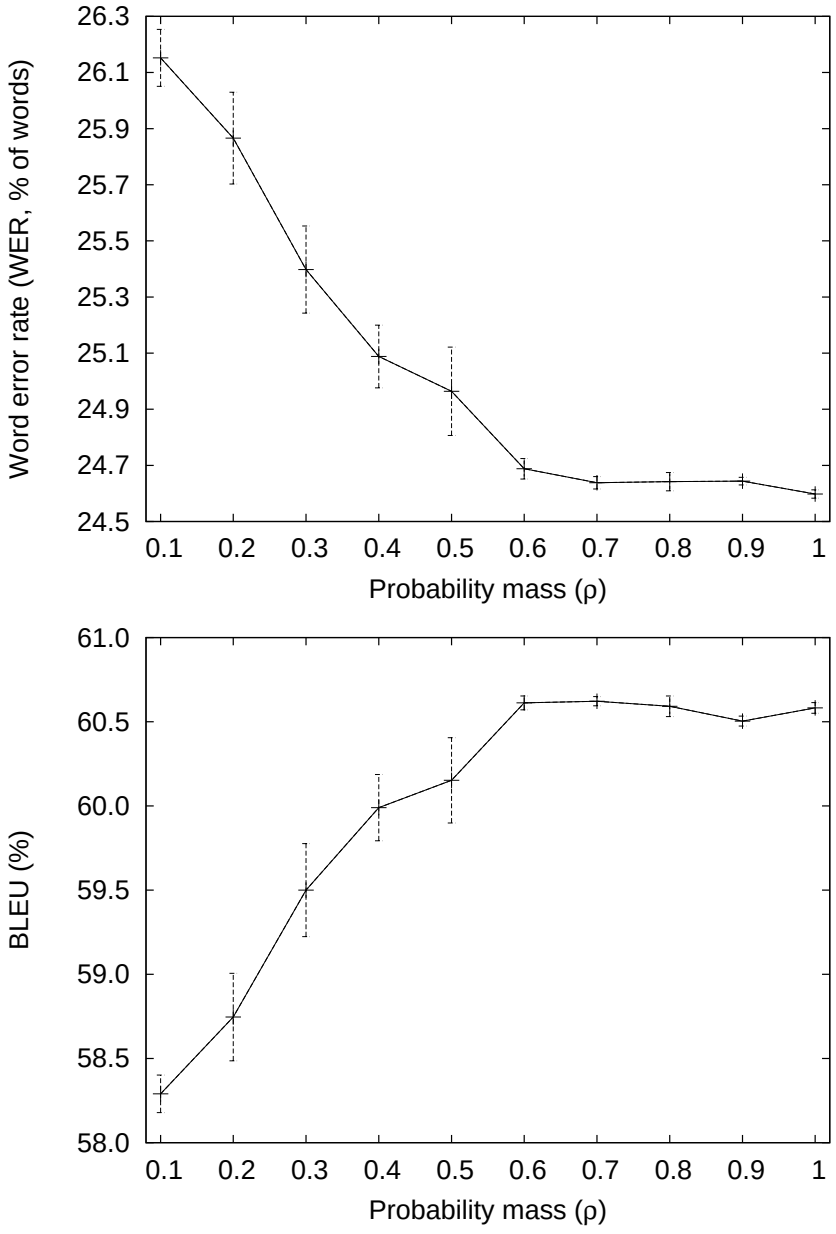


Figure 3.2: For the different values of ρ used, mean and standard deviation of the WER (top) and the BLEU score (bottom) achieved after training the French PoS tagger, Catalan being the TL.

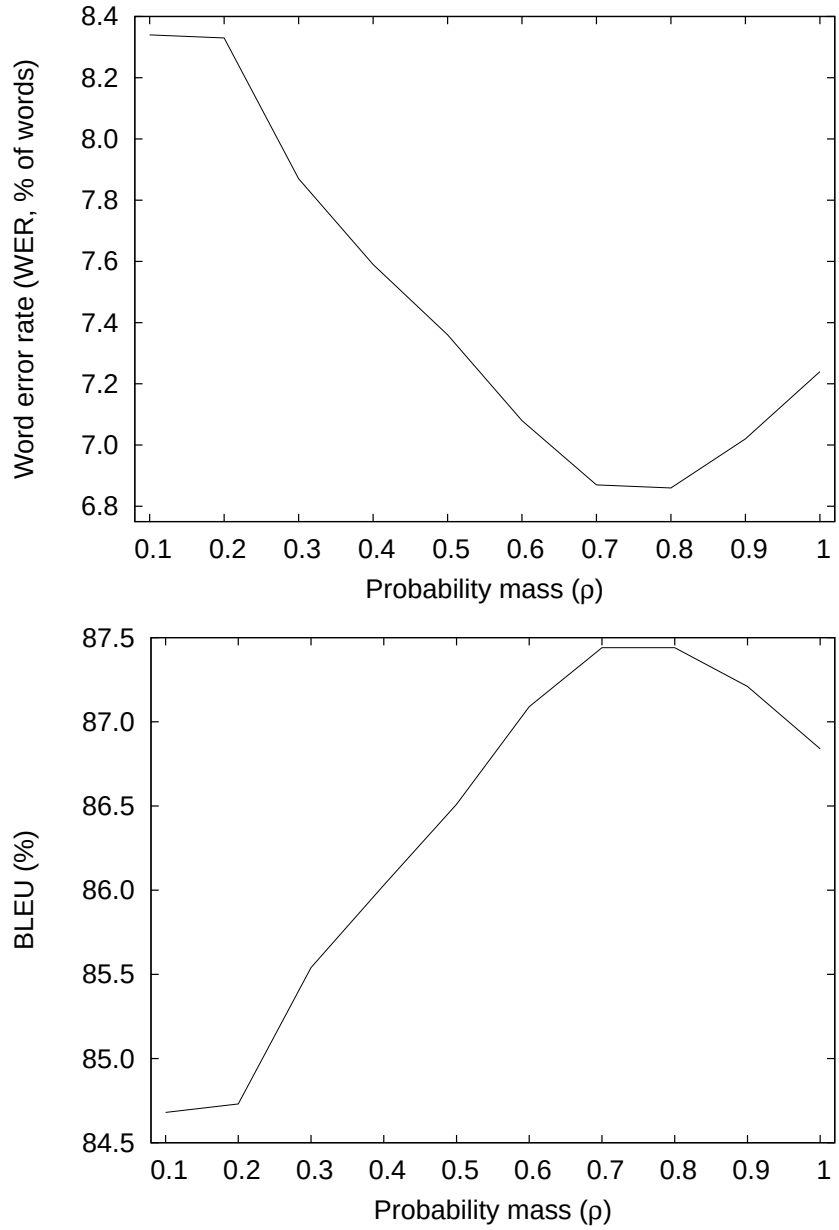


Figure 3.3: For the different values of ρ used, WER (top) and BLEU score (bottom) achieved after training the Occitan PoS tagger, Catalan being the TL.

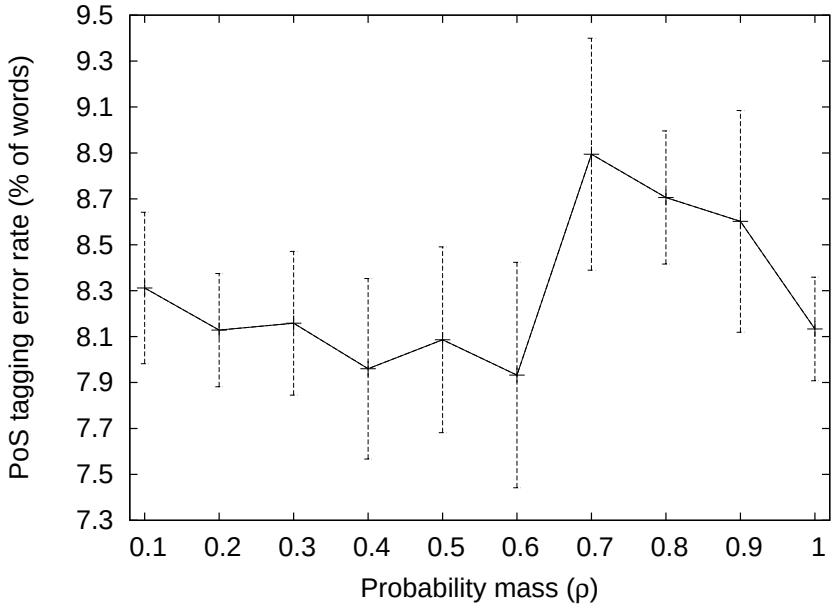


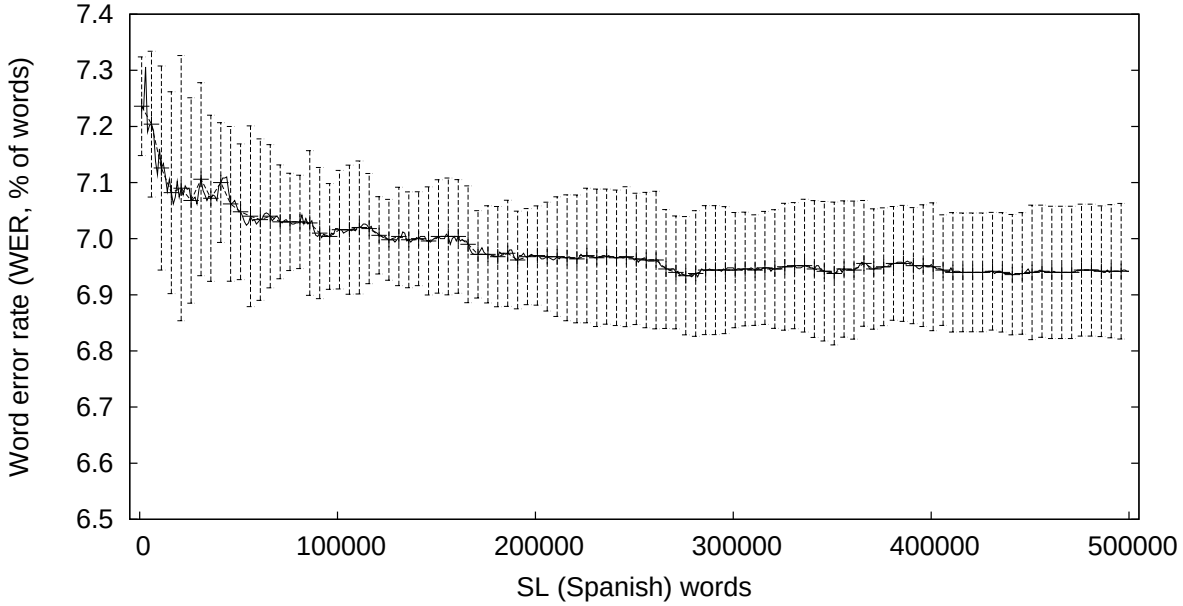
Figure 3.4: For the different values of ρ used, mean and standard deviation of the PoS tagging error rate of the Spanish PoS tagger after processing the whole training corpus.

method; as a consequence of this linear combination, in this case, the curve has a steeper slope, since the initial model is not as good as the model being learned.

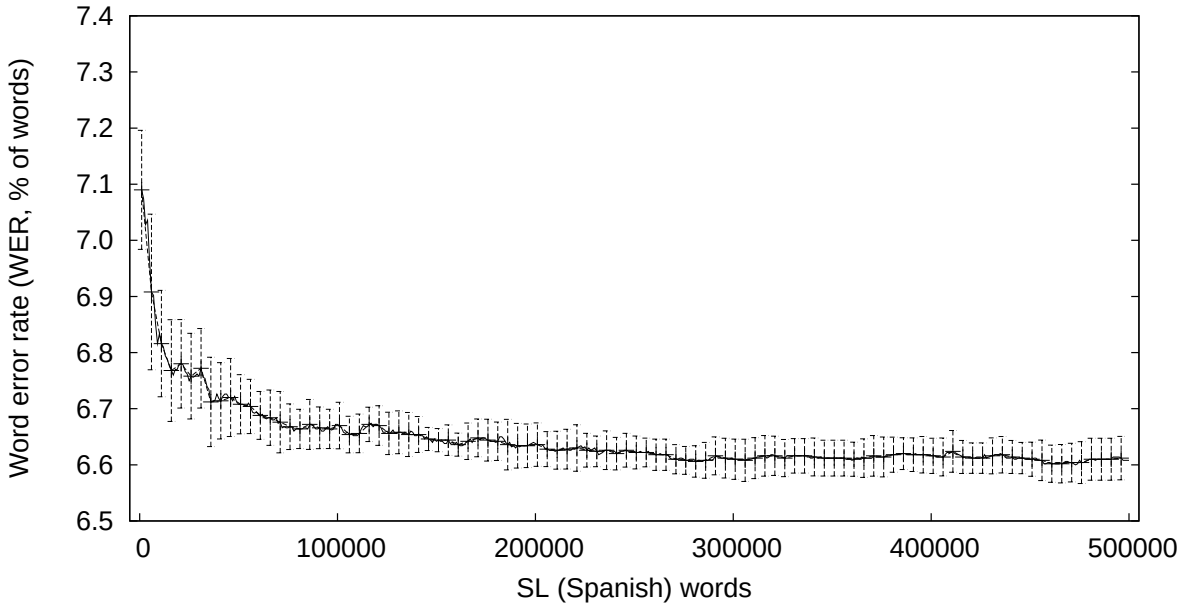
Finally, as to how many translations are avoided with the proposed pruning method, Figure 3.8 shows, for the three languages being studied, the average ratio and standard deviation of the number of words finally translated with respect to the total number of words translated when no pruning is performed. As can be seen, for the values of ρ that produce the most accurate PoS tagger to be used in MT (0.9 for Spanish, 0.7 for French, and 0.8 for Occitan) the percentage of words translated is around 20%. This percentage can be taken to be roughly proportional to the percentage of disambiguation paths needed to reach the corresponding mass probability threshold.

3.4 Discussion

In order to overcome the main disadvantage of the MT-oriented training method, the huge number of translations to perform by the training algorithm, a disambiguation path pruning technique based on *a priori* knowledge, obtained in an unsupervised way from the SL, has been proposed and tested. This pruning method is based on the assumption that any reasonable model of SL tags may prove helpful to choose a set of possible disambiguation paths, the correct one being included in that set. Moreover, the model used for pruning can be updated during training with the new data collected while training.

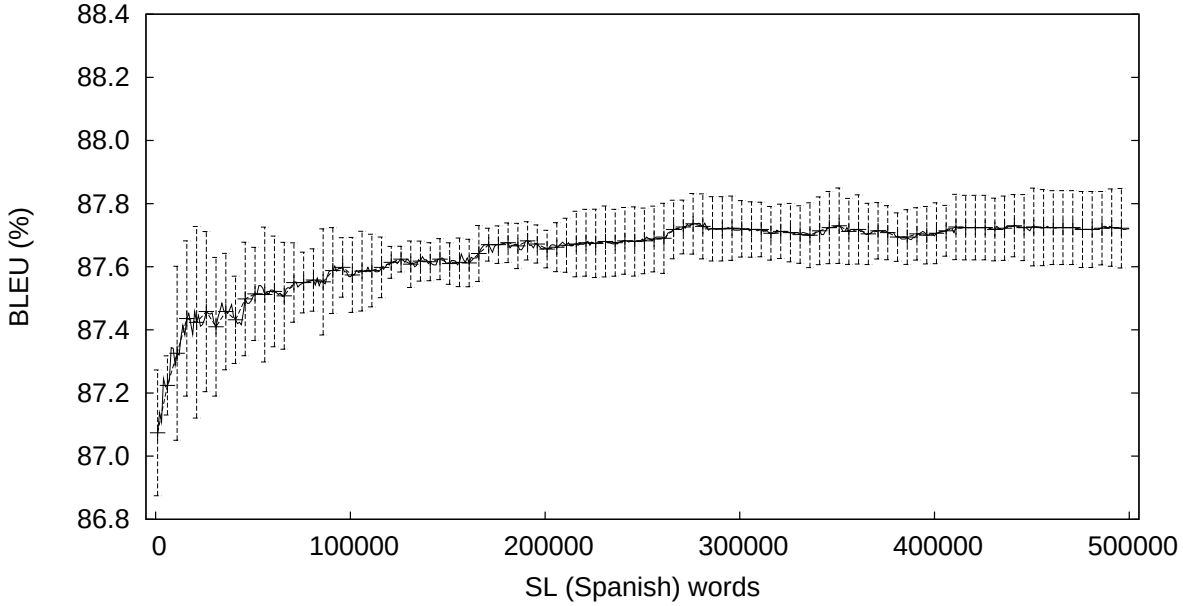


(a) $\rho = 0.1$

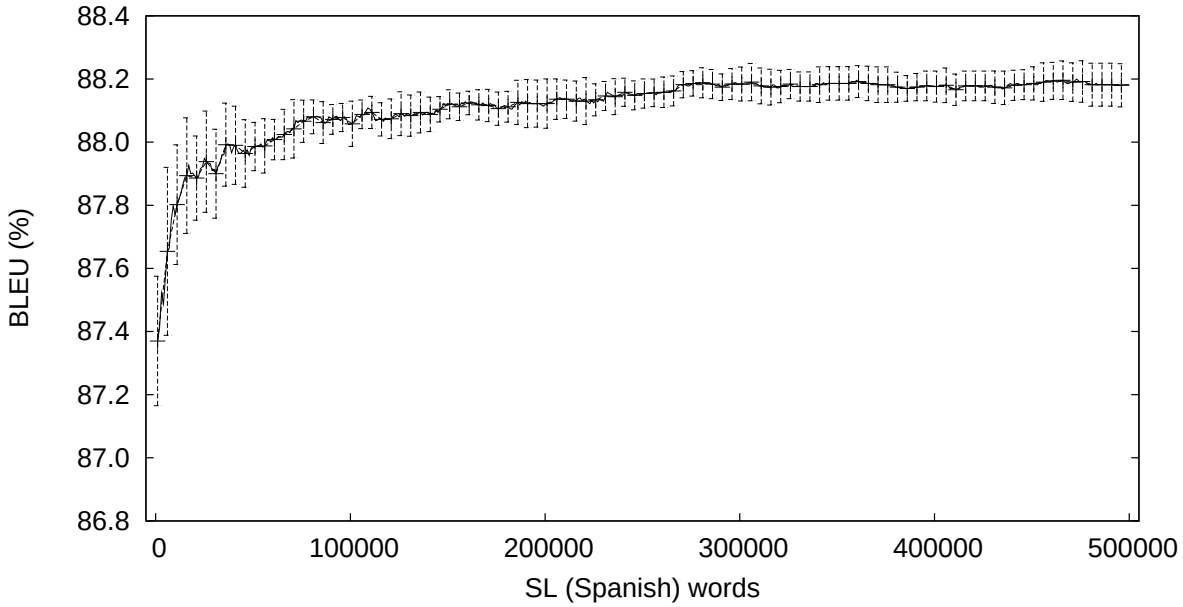


(b) $\rho = 0.9$

Figure 3.5: Evolution of the mean and standard deviation of the WER for two different values of the probability mass threshold ρ when training the Spanish PoS tagger. The other two languages behave in a similar way.



(a) $\rho = 0.1$



(b) $\rho = 0.9$

Figure 3.6: Evolution of the mean and standard deviation of the BLEU score for two different values of the probability mass threshold ρ when training the Spanish PoS tagger. The other two languages behave in a similar way.

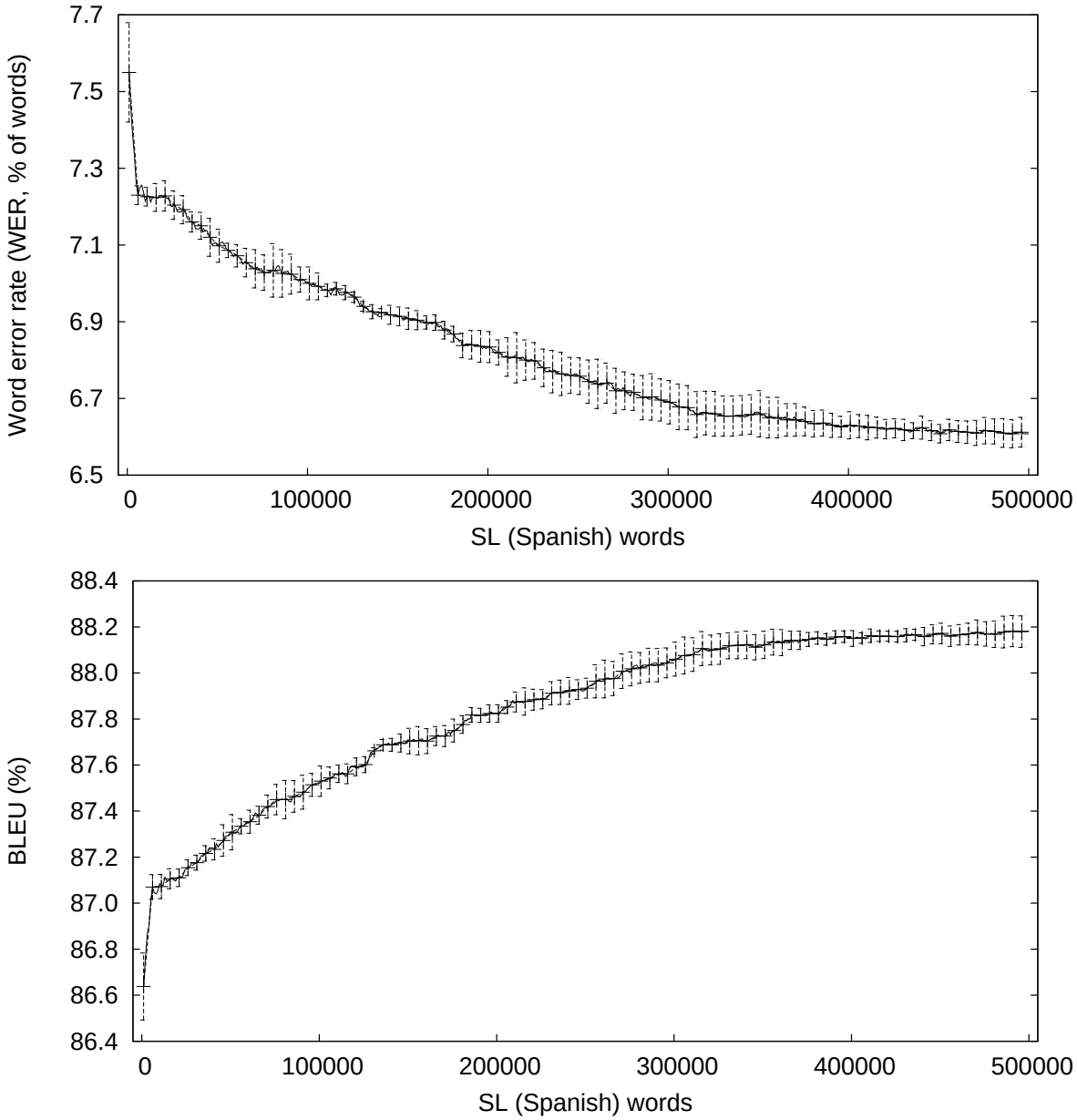


Figure 3.7: Evolution of the mean and the standard deviation of the WER (top) and the BLEU score (bottom) of the Spanish-to-Catalan MT system when embedding the model used for pruning when training the Spanish PoS tagger for $\rho = 0.9$. This model is a linear combination of the initial model and the model being trained.

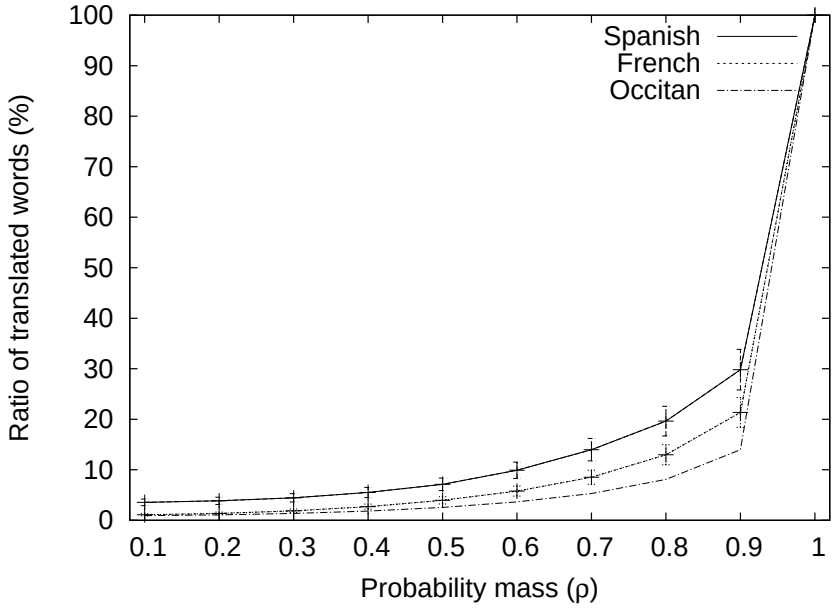


Figure 3.8: For each language, percentage of translated words for each value of the probability mass threshold ρ. The percentage of translated words is calculated over the total number of words that are translated when no pruning is done.

The results show, on the one hand, that the pruning method described avoids having to perform around 80% of the translations; and on the other hand, that the translation quality achieved is not affected when improbable disambiguation paths are not taken into account, and may even be slightly improved (a reduction in WER of 0.4 percentage points was reported for Occitan-to-Catalan translation). It is worth noting, however, that the PoS tagging performance shows erratic behaviour. This demonstrates further that good translations may be achieved independently of the actual PoS tagging performance.

Finally, it can be concluded that a value of ρ around 0.8 is enough to speed up the MT-oriented training method and, at the same time, to ensure good translation performance of the resulting PoS tagger when building new PoS taggers to be used in RBMT.

Chapter 4

Part-of-speech tag clustering

In previous chapters, the tagset (that is, the set of hidden states of the Markov process) was fixed beforehand following linguistic criteria; however, the automatic inference of such set of states is possible. This chapter focuses on the automatic inference of the tagset to be used by PoS taggers involved in RBMT. To that end, a bottom-up agglomerative clustering algorithm is applied over the states of an initial HMM using the fine-grained PoS tags delivered by the morphological analyzer of the MT system; this initial HMM is trained by means of the MT-oriented method described in Chapter 2.

4.1 Motivation

The reason for reducing the number of tags (states) used by HMM-based PoS taggers is due to the fact that the fewer tags the tagset has, the better the HMM parameters are estimated, thanks to the reduction of the data sparseness problem. Furthermore, as the number of transition probabilities to estimate grows (for a first-order HMM, quadratically with the number of tags), the number of parameters to store (and retrieve) may be drastically reduced. This may be a desirable effect if the HMM-based PoS taggers are involved in online tasks, such as online MT, because they may start up faster and use less memory.

The reduction of the data spareness problem may cause a slight improvement in the tagging performance as described by Brants (1995a,b); experiments will show whether that small improvement also arises when the tagset is inferred through bottom-up agglomerative clustering and whether it also affects translation quality.

4.2 Clustering algorithm

As it was introduced in Chapter 1, the model merging algorithm cannot be applied mainly because it needs a hand-tagged corpus to compute the initial model. Instead, a *bottom-up agglomerative clustering* (Duda et al., 2001, p. 552; Webb, 2002, p. 368) is applied over the states of an initial HMM that has as many states as different fine-grained PoS tags the morphological analyzer delivers (see Section 4.5 for details about the different fine-grained PoS tags delivered by the morphological analyzer).

Bottom-up (hierarchical) agglomerative clustering is a general clustering algorithm that starts with as many clusters as different objects (in our case, there will be one cluster per fine-grained PoS tag) and in each step it selects the two clusters that are closer according to a similarity measure and merges them into a single one. The algorithm stops when there are no more clusters to merge or when the dissimilarity between clusters is higher than a given threshold. Agglomerative clustering has the advantage, over other clustering algorithms such as K-means, in that it automatically determines the final number of clusters.

4.3 Constraint on the clustering

A very important property of the resulting tagset is that it must be possible to restore the original information (all grammatical features) represented by the fine-grained PoS tags from the coarser ones; note that this is the information we are interested in, as it is used by the subsequent MT modules to carry out the translation. To ensure this property a constraint must hold; this constraint, already used by Brants (1995b), establishes that two tags (states) cannot be merged in the same cluster if they share the emission of one or more observable outputs (word classes). This is because in this case, the PoS tagger would not be able to choose a PoS tag for that observable output, that is, it would not be able to disambiguate that word.

The following example illustrates the necessity for this constraint: consider the English word *houses* which may be tagged as a verb with present tense, third person, plural or as a noun in plural; if the clustering algorithm decides to merge both fine-grained PoS tags into a single cluster it becomes clear that the PoS tagger would not be able to assign one of the two fine-grained PoS tags to the word *houses* when tagging an input text, since both tags would be represented by the same HMM state (cluster).

The previous constraint can be formally described as follows. Let γ^f be a fine-grained PoS tag, γ^c a coarse tag (cluster), σ an observable output, and Γ^f, Γ^c and Σ the fine-grained tagset, the coarse one and the set of observable outputs, respectively; the coarse tagset Γ^c constitutes a partition of the fine-grained tagset Γ^f, that is, $\Gamma^c \in \mathcal{P}(\Gamma^f)$. The original information of the fine-grained PoS tag γ^f can be retrieved from the coarse

one γ^c by means of the injective function h defined as:

$$h : \Sigma \times \Gamma^c \rightarrow \Gamma^f \tag{4.1}$$

To ensure that this function is injective, that is, that for a given observable σ and a given coarse tag γ^c there is only one fine-grained PoS tag γ^f, the next constraint must be met:

$$\forall \gamma^c \in \Gamma^c, \sigma \in \Sigma, \gamma_i^f, \gamma_j^f \in \gamma^c, \gamma_i^f \neq \gamma_j^f : \gamma_i^f \in \sigma \Rightarrow \gamma_j^f \notin \sigma, \tag{4.2}$$

where with $\gamma^f \in \gamma^c$ we mean that the fine-grained PoS tag γ^f is in the cluster (coarse tag) denoted by γ^c, and with $\gamma^f \in \sigma$ we mean that the observable output σ can be emitted from the fine-grained PoS tag γ^f.

If the constraint expressed in (4.2) holds, function h is injective, and no information is lost when grouping fine-grained PoS tags into coarser ones.

4.4 Distance between clusters

In order to measure how similar (dissimilar) two clusters are, a distance between them is defined. The smaller the distance between two clusters, the more similar they are. Before defining how the distance between two clusters is calculated, the distance between two fine-grained PoS tags (states) must be defined, as the former is based on the latter.

The distance between two fine-grained PoS tags is based on the Kullback-Leibler *directed logarithmic divergence* (Kullback and Leibler, 1951) applied to the probabilistic distributions defined by the transition probabilities A between each fine-grained PoS tag (state) and the rest. The directed logarithmic divergence measures the relative entropy between two probabilistic distributions:

$$d(\gamma_i^f, \gamma_j^f) = \sum_{\gamma_k^f \in \Gamma^f} a_{\gamma_i^f \gamma_k^f} \log_2 \frac{a_{\gamma_i^f \gamma_k^f}}{a_{\gamma_j^f \gamma_k^f}}. \tag{4.3}$$

Since $d(\gamma_i^f, \gamma_j^f) \neq d(\gamma_j^f, \gamma_i^f)$, the relative entropy is not a true metric, but it satisfies some important mathematical properties; it is always nonnegative and equals zero only if $\forall \gamma_k^f \; a_{\gamma_i^f \gamma_k^f} = a_{\gamma_j^f \gamma_k^f}$.

A symmetric distance measure is needed for the clustering algorithm; in the experiments in Section 4.5, the *intrinsic discrepancy* (Bernardo and Rueda, 2002) has been used, and is defined as:

$$\delta(\gamma_i^f, \gamma_j^f) = \min(d(\gamma_i^f, \gamma_j^f), d(\gamma_j^f, \gamma_i^f)). \tag{4.4}$$

This is always finite, even when one of the probabilistic distributions has null values for some range of X, given that the other probabilistic distribution does not.

The distance between two clusters is defined as:

$$\delta(\gamma_i^c, \gamma_j^c) = \begin{cases} \frac{\sum_{\gamma_k^f \in \gamma_i^c} \sum_{\gamma_l^f \in \gamma_j^c} \delta(\gamma_k^f, \gamma_l^f)}{|\gamma_i^c||\gamma_j^c|} & \text{if } \delta(\gamma_k^f, \gamma_l^f) \leq \rho \ \forall \gamma_k^f \in \gamma_i^c, \gamma_l^f \in \gamma_j^c \\ \infty & \text{otherwise} \end{cases} \quad (4.5)$$

where ρ is the distance threshold. This distance is analogous to the *pair-group average* distance (Duda et al., 2001, p. 553), but in this case it is made equal to infinity if there are at least a couple of fine-grained PoS tags that are more dissimilar than the given threshold ρ. This prevents the clustering algorithm from putting very heterogeneous (in terms of their transition probabilities) fine-grained PoS tags in the same cluster.

4.5 Experiments

This section reports the experiments conducted to test the approach presented in this chapter to automatically infer the tagset to be used by the HMM-based PoS tagger involved in RBMT.

4.5.1 Task and evaluation

As in the previous two chapters, this section focuses on the same three languages —Spanish, French and Occitan— all being translated into Catalan by means of the open-source shallow-transfer MT platform Apertium (see appendix A).

The experiments conducted consist of training an initial HMM-based PoS tagger using the fine-grained PoS tags by means of the MT-oriented method described in Chapter 2, and then applying the clustering algorithm described in Section 4.2 over the states of that initial model. To train the initial model the same linguistic data and training corpora used in previous chapters (see Section 2.5.1) are used.

Table 4.1 shows for each language the number $|\Gamma^f|$ of fine-grained PoS tags, the number $|\Sigma|$ of ambiguity classes (word classes) of the initial HMM, the number of fine-grained PoS tags that correspond to single words and the number of them that correspond to multi-word expressions.[1] As can be observed, the number of fine-grained PoS tags due to multi-word expressions is similar for Spanish and Occitan, while it is considerably smaller in the case of French. This is explained by the fact that most of the multi-word expressions correspond to verbs with attached enclitic pronouns; in Spanish and Occitan one (Spanish *dame* = "give+me") or two (Spanish *dámelo* = "give+me+it") enclitic pronouns can be attached to verbs in infinitive, present participle or imperative tense; however, in French clitic pronouns can only be attached to verbs in imperative mood.

[1] This data, which has been already reported in Table 2.1 on page 27, is repeated here for convenience.

Language	$\|\Gamma^f\|$	$\|\Sigma\|$	single-word	multi-word
Spanish	2 116	3 061	377	1 739
French	422	873	320	102
Occitan	2 305	3 809	348	1 957

Table 4.1: Number of fine-grained PoS tags $|\Gamma^f|$, number of ambiguity classes (word classes) $|\Sigma|$, number of fine-grained PoS tags that correspond to single words delivered by the morphological analyzer, and number of them that correspond to multi-word expressions.

In order to find the threshold that produces the best tagset, the bottom-up agglomerative clustering method was applied for thresholds varying from 0 to 2.5 in increments of 0.05. Furthermore, to evaluate the effect of the clustering on the translation quality and the tagging performance (only for Spanish) the same test corpora and references already used in previous chapters were used. Note that the evaluation of the tagging performance is slightly different because in this experiment unknown words are not treated as ambiguous words that can be assigned a category in the set of *open* categories (as in previous chapters, see Section B.1 in appendix B), but as unambiguous words receiving a unique PoS tag that identifies all unknown words. Proceeding in this way, the clustering is not restricted by the fact that all HMM states for PoS tags corresponding to open-class words would share the emission of the *open* word class. Recall from Section 4.3 that two states cannot be merged if they share the emission of one or more word classes.

4.5.2 Results

Figure 4.1 shows, on the one hand, the mean and the standard deviation of the WER and BLEU score achieved when translating Spanish into Catalan, for the different threshold values and, on the other hand, the mean number of coarse tags (clusters) obtained with each threshold value. Note that the smallest value (a negative threshold $\rho < 0$) corresponds to the initial HMM (before clustering). Also note that after applying the clustering algorithm, the HMM parameters are recomputed using the fractional counts collected during training; this would be equivalent to retraining with the new tagset. Therefore, there is no need to retrain the HMM for each inferred tagset; one simply recalculates the transition and emission probabilities.

As can be seen in Figure 4.1 the use of a clustering algorithm drastically reduces the number of states of the HMM. Note that with a null threshold the number of states is around 375, that is, there are about 1 700 fine-grained PoS tags that have exactly the same transition probabilities. This is because these fine-grained PoS tags are mostly for verbs receiving one or two enclitic pronouns, which rarely appear in the training corpus; therefore, the clustering algorithm puts all of them in the same

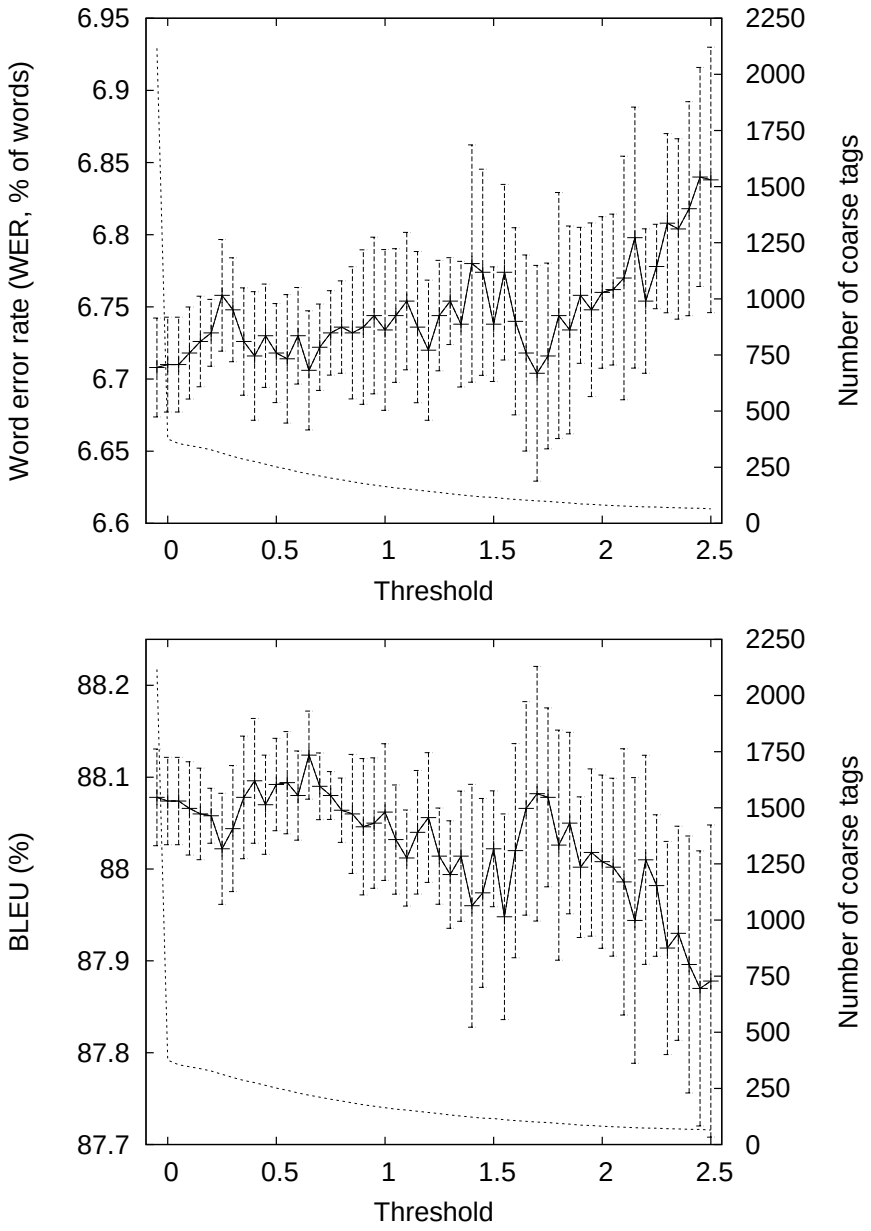


Figure 4.1: Mean and the standard deviation of the WER (top) and the BLEU score (bottom) when translating Spanish into Catalan for the different threshold values used to automatically infer the set of states to be used by the Spanish PoS tagger. The mean number of tags of the obtained tagset for each threshold is also given (dotted line with values on the right vertical axis).

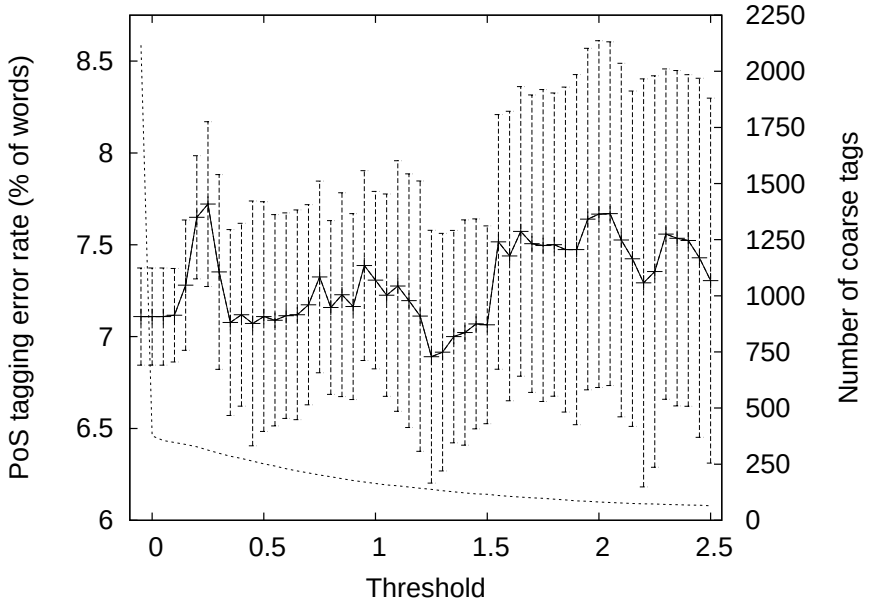


Figure 4.2: Mean and standard deviation of the PoS tagging error rate achieved for the different threshold values used to automatically infer the set of states to be used by the Spanish PoS tagger. The mean number of tags of the inferred tagset for each threshold is also given (dotted line with values on the right vertical axis).

cluster. Furthermore, the translation quality fluctuates for threshold values between 0 and 1.7 and it is not improved in this range compared with that of using the fine-grained PoS tagset. Moreover, the standard deviation is smaller in the case of a null threshold, compared with that of the remaining threshold values, which indicates that the described behaviour is not uniform across all the training corpora; for instance, with some of the training corpora used to estimate the initial HMM there is an improvement in the translation quality, while in others there is no improvement or the translation quality even becomes worse. In any case, translation quality is not seriously affected; note that WERs vary from 6.71 to 6.77 for threshold values between 0 and 1.7.

Concerning the PoS tagging performance, Figure 4.2 shows the mean and standard deviation of the PoS tagging error rate for the different threshold values used. As in Figure 4.1, the number of coarse tags of the inferred tagset is also provided. Note that the PoS tagging error rates reported cannot be directly compared to those provided in previous chapters, because in this evaluation unknown words are not taken into account for the computation of the PoS tagging error rate; recall from the end of Section 4.5.1 that unknown words are treated as unambiguous in all the experiments conducted in this chapter.

Figure 4.2 shows a small improvement in the PoS tagging performance for a threshold value $\rho = 1.25$. Note that for this threshold value there is no improvement at all in the translation quality; moreover, the best translation quality is achieved for a

threshold value of 1.7, for which the PoS tagging accuracy is close to the worst achieved with the different thresholds tested. This result supports the underlying assumption made in this thesis: that good translation quality does not necessary imply good PoS tagging performance. Concerning the standard deviation, it becomes bigger as the threshold value increases; this behaviour indicates that the PoS error rate greatly differs depending on the corpus used to train the initial model that uses the fine-grained PoS tags.

Figure 4.3 shows, on the one hand, the mean and the standard deviation of the WER and the BLEU score for the French-to-Catalan translation with the different threshold values and, on the other hand, the mean number of coarse tags (clusters) obtained with each threshold value. Note that, as in the case of Spanish, the negative threshold corresponds to the initial HMM. In that figure we can see that the translation quality is worse than that of the initial HMM for all threshold values tested, and that the number of clusters for a null threshold value is close to the number of fine-grained PoS tags that correspond to single words, as already shown in the case of Spanish.

Finally, Figure 4.4 shows the WER and the BLEU score of the Occitan-to-Catalan translation achieved for the different threshold values tested. As can be seen, the translation quality is again worse than that of the initial HMM for all threshold values tested; however, for small thresholds such as 0.25, the translation quality is not seriously affected and at the same time the number of tags is drastically reduced from 2305 to 339 tags. It must be said that when a null threshold value is used the number of tags is 389; as in Spanish and French, those fine-grained PoS tags that do not appear in the training corpus have the same transition probabilities and the clustering algorithm puts all of them in the same cluster.

4.6 Discussion

This chapter has explored automatic tagset reduction, starting from a large fine-grained tagset, by means of a bottom-up agglomerative clustering algorithm. The results reported show that this strategy is not as good as initially expected. For the three languages studied, all being translated into Catalan, there is no improvement in the translation quality, as evaluated against human-corrected translations. However, in the case of Spanish the translation quality is not seriously affected and at the same time the number of states is drastically reduced. The results demonstrate that the tagset has little impact on the translation quality achieved, provided that the constraint on the clustering is met; therefore, the presented strategy may be adequate to infer small tagsets for HMM-based PoS taggers to be used in RBMT systems involved in tasks, such as online MT, in which a reduced consumption of memory and startup time is desirable.

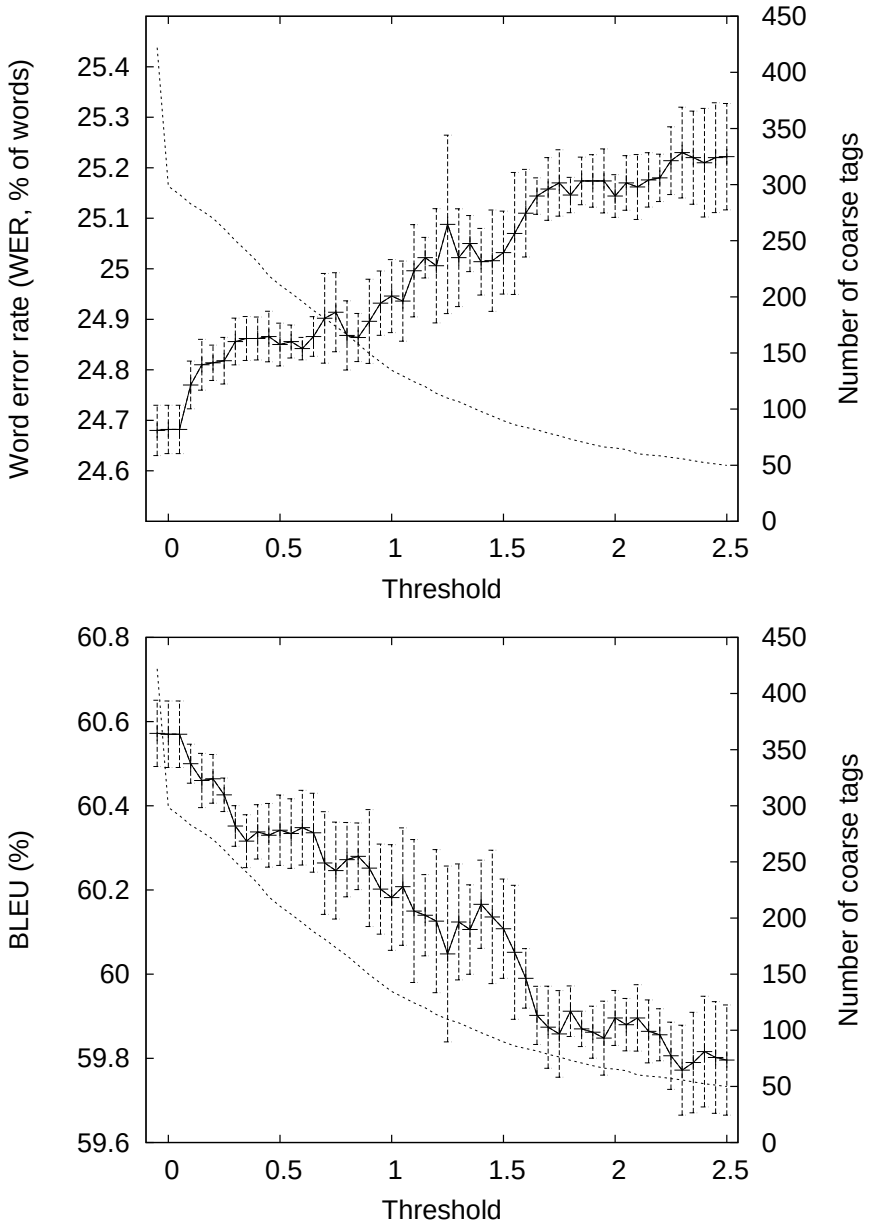


Figure 4.3: Mean and the standard deviation of the WER (top) and the BLEU score (bottom) when translating French into Catalan for the different threshold values used to automatically infer the set of states to be used by the French PoS tagger. The mean number of tags of the inferred tagset for each threshold is also given (dotted line with values on the right vertical axis).

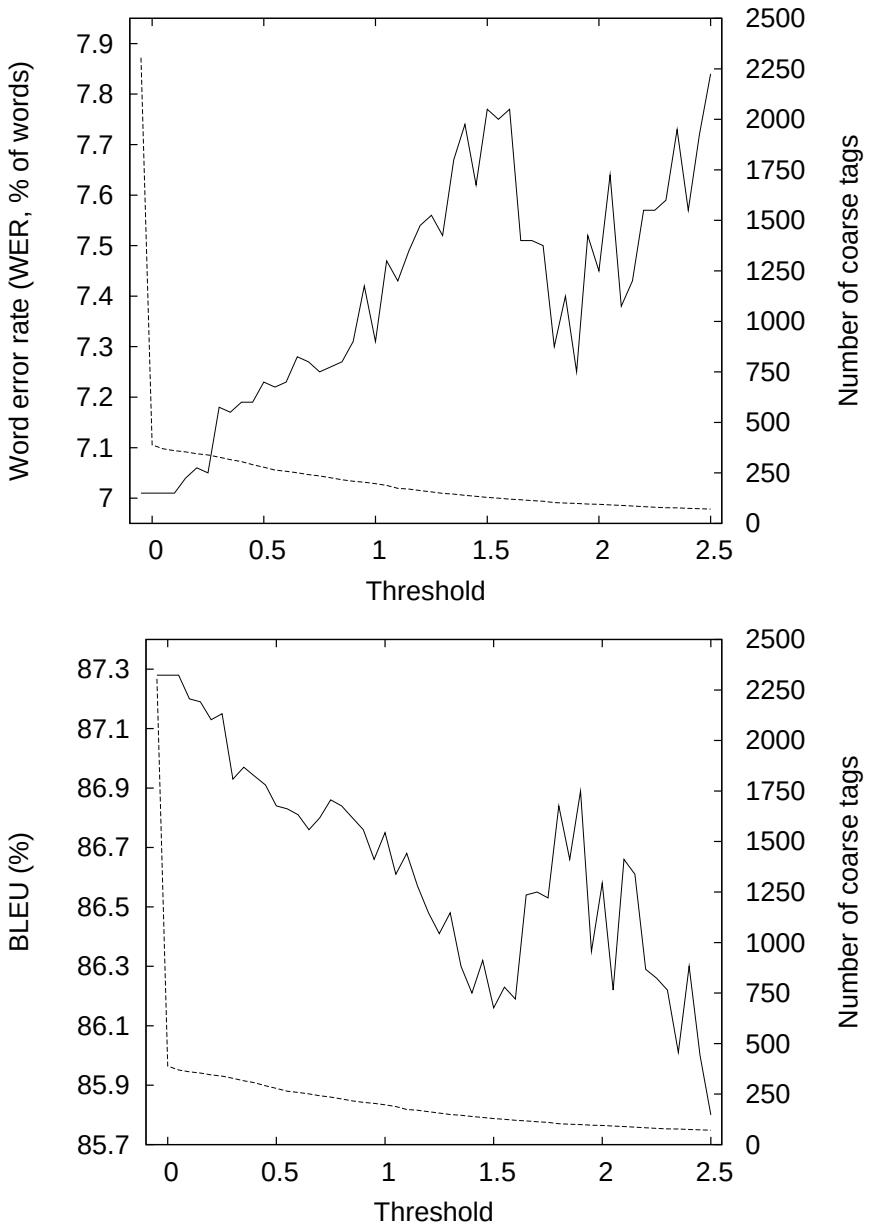


Figure 4.4: WER (top) and the BLEU score (bottom) of the Occitan-to-Catalan translation for the different threshold values used to automatically infer the set of states to be used by the Occitan PoS tagger. The number of tags of the inferred tagset for each threshold is also given (dotted line with values on the right vertical axis).

The presented strategy does not provide the expected results; this may be due to how the distance between two clusters is calculated. As the distance measure used is analogous to the *pair-group average* the clustering algorithm may put in the same cluster very different fine-grained PoS tags as a consequence of what may be called the "average" effect. Better results may be expected if the HMM parameters were recalculated after each merging, as the distance between two clusters would be just the intrinsic discrepancy between their transition probabilities. The main disadvantage of this approach is that the intrinsic discrepancy between each pair of clusters needs to be recalculated for all of them after each merging, making the clustering algorithm much slower.

Chapter 5

Automatic inference of shallow-transfer rules for machine translation

This chapter focuses on the inference of structural transfer rules for shallow-transfer MT. Transfer rules are generated from alignment templates (ATs), like those used in SMT, that have been extracted from parallel corpora and extended with a set of restrictions that control their application. The experiments conducted show an improvement in the translation quality, as compared to word-for-word translation (when no transfer rules are used), and that the translation quality achieved is close to that obtained when using hand-coded transfer rules. The method in this chapter is entirely unsupervised and benefits from information in the rest of modules of the RBMT system in which the inferred rules are applied.

5.1 Introduction

As was mentioned in the introductory chapter, building an RBMT system requires a huge human effort. Previous chapters have focused on easing the development of PoS taggers to be used in RBMT; this chapter presents an unsupervised method aimed at easing the development of an RBMT system by inferring from a small amount of parallel text the transfer rules (in this case, shallow-transfer rules) to be used in translation.

The method in this chapter works by adapting the alignment template (AT) approach (Och, 2002; Och and Ney, 2004) introduced in the SMT framework to the rule-based approach. To that end:

- the bilingual dictionary of the MT system in which the inferred transfer rules will be integrated is used to ensure that the lexical content of each bilingual phrase[1] pair extracted from the training corpus (see Section 5.2.2) can be reproduced by the MT system;
- linguistically motivated word classes are used to generalize the extracted bilingual phrases, deriving ATs from them; and,
- a set of restrictions is attached to each AT to control its application as part of a transfer rule; therefore, extending the definition of ATs.

Once these extended ATs have been extracted from the training corpora, transfer rules are generated from them. In the experiments reported in Section 5.5, shallow-transfer rules to be used by the Apertium MT engine (see appendix A) are generated directly in Apertium's XML-based transfer language.

5.2 The alignment template approach

The alignment template (AT) approach (Och, 2002; Och and Ney, 2004) was introduced in the SMT framework as a feature function for the log-linear maximum entropy model (Och and Ney, 2002) to generalize the knowledge learned for a specific phrase to similar phrases.

An AT performs a generalization over bilingual phrase pairs using word classes instead of words. An AT $z = (S_m, T_n, G)$ consists of a sequence S_m of m SL word classes, a sequence T_n of n TL word classes, and a set of pairs $G = \{(i, j) : i \in [1, n] \wedge j \in [1, m]\}$ with the alignment information between the TL and SL word classes in the two sequences.

Learning a set of ATs from a sentence-aligned parallel corpus consists of:

1. the computation of the word alignments,
2. the extraction of bilingual phrase pairs, and
3. the generalization of such bilingual phrase pairs by using word classes instead of the words themselves.

5.2.1 Word alignments

A variety of methods, statistical (Och and Ney, 2003) or heuristic (Caseli et al., 2005), may be used to compute word alignments from a (sentence-aligned) parallel corpus.

[1]For the purpose of this chapter, *phrase* means any sequence of consecutive words, not necessarily whole syntactic constituents.

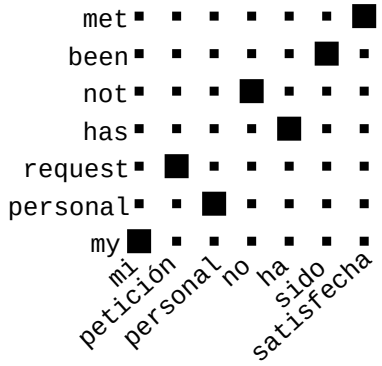


Figure 5.1: Example of word-aligned Spanish–English sentence pair. The alignment information is represented as a binary matrix.

In the experiments reported in Section 5.5, word alignments are obtained by training classical statistical translation models to translate from language L_1 to language L_2 (and vice versa) and then computing the Viterbi alignments under the previously estimated translation models. The Viterbi alignment between SL and TL sentences is defined as the alignment whose probability is maximal under the translation models previously estimated. The way in which Viterbi alignments are obtained from all the translation models used is as follows:

- first, the Viterbi alignment of each sentence pair is calculated using the simplest alignment model,
- then this alignment is iteratively improved with respect to the alignment probability of the refined translation models.

Once the Viterbi alignments are computed, the resulting alignments G_1 and G_2 (one for each translation direction) are symmetrized through the refined intersection method proposed by Och and Ney (2003, p. 33).

Figure 5.1 shows a Spanish–English sentence pair and the alignment between their words. The alignment information is represented as a binary matrix in which a value of 1 (black squares) means that the words at the corresponding positions are aligned; analogously, a value of 0 means that the words are not aligned.

Training

In order to train the translation models and to calculate the Viterbi alignments of each pair of aligned sentences found in the training corpus the open-source `GIZA++` toolkit[2] (Och and Ney, 2003) is used.

[2] `http://code.google.com/p/giza-pp`

The training of the word alignments consists of:

1. training the IBM model 1 (Brown et al., 1993) for 5 iterations; in this model, word order does not affect the alignment probabilities;

2. training the HMM alignment model (Vogel et al., 1996) for 5 iterations; this alignment model has the property of making alignment probabilities explicitly dependent on the alignment position of the previous word;

3. training the IBM model 3 (Brown et al., 1993) for 5 iterations; in this model, the probability of an alignment depends on the positions of the aligned words and on the length of SL and TL sentences; and

4. training the IBM model 4 (Brown et al., 1993) for 5 iterations; in this model, one has the same dependency between the positions of the aligned words as in model 3 and two additional dependencies: one on which SL and TL words are aligned, and another one on the position of any other SL words being aligned with the same TL word.

Note that after obtaining the Viterbi alignments these statistical translation models are no longer used.

5.2.2 Extraction of bilingual phrase pairs

Bilingual phrase pairs are automatically extracted from the word-aligned sentence pairs; the extraction of bilingual phrase pairs (Zens et al., 2002) is performed by considering all possible pairs below a certain length and ensuring that:

1. all words are consecutive, and

2. words within the bilingual phrase pair are not aligned with words from outside.

The set $\mathrm{BP}(w_{\mathrm{S}1}^{J}, w_{\mathrm{T}1}^{I}, G)$ of bilingual phrases that are extracted from the word-aligned sentence pair $(w_{\mathrm{S}1}, \ldots, w_{\mathrm{S}J}), (w_{\mathrm{T}1}, \ldots, w_{\mathrm{T}I})$ may be formally expressed as follows:

$$\begin{aligned}\mathrm{BP}(w_{\mathrm{S}1}^{J}, w_{\mathrm{T}1}^{I}, G) = \{(w_{\mathrm{S}j}^{j+m}, w_{\mathrm{T}i}^{i+n}) : \forall (i', j') \in G : \\ j \leq j' \leq j+m \Leftrightarrow i \leq i' \leq i+n\}.\end{aligned} \tag{5.1}$$

Figure 5.2 shows the set of bilingual phrase pairs with more than one SL word extracted from the word-aligned Spanish–English sentence pair shown in Figure 5.1.

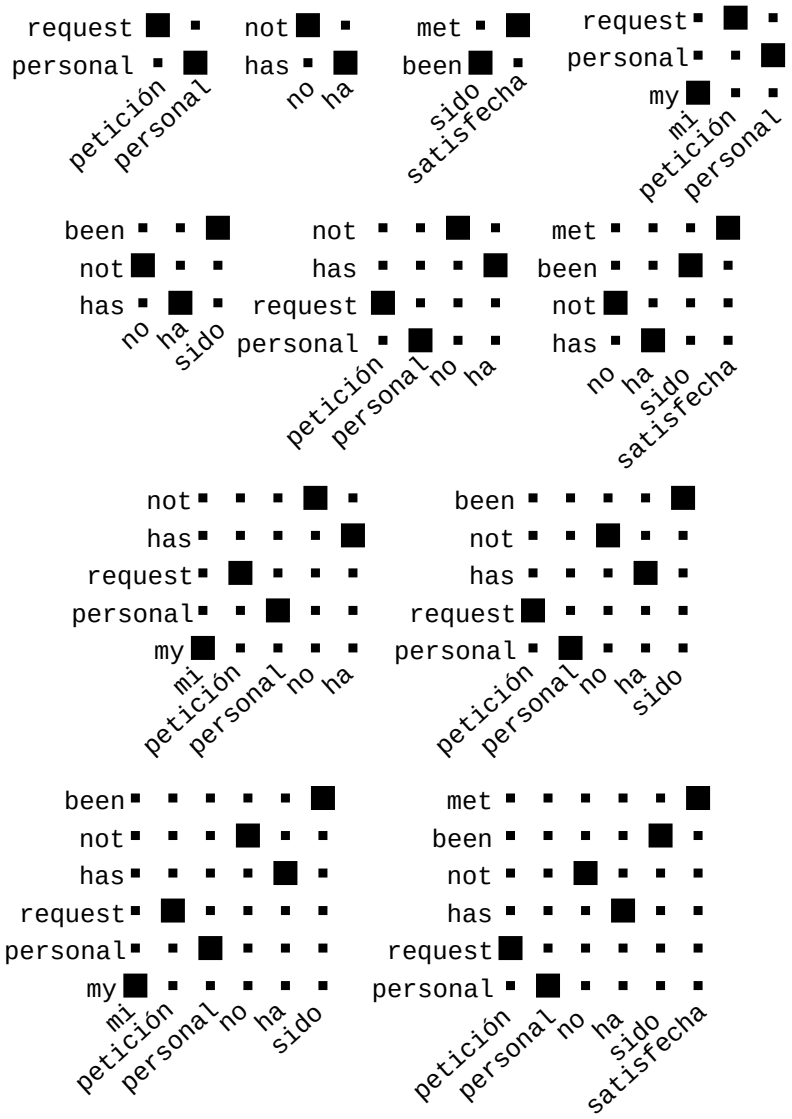


Figure 5.2: Set of bilingual phrase pairs (see Section 5.2) extracted from the word-aligned Spanish–English sentence pair shown in Figure 5.1. Note that bilingual phrase pairs containing only one word and the whole word-aligned sentence pair have been omitted.

5.2.3 Generalization

The generalization of the bilingual phrase pairs is simply done by using word classes instead of the words themselves; to that end a function $C(\cdot)$ that maps single words into word classes is defined. The use of word classes allows the description of word reorderings, preposition changes and other divergences between SL and TL. Och and Ney (2004) use automatically obtained word classes (Och, 1999) to extract ATs for SMT. However, for RBMT, linguistically motivated word classes must be used (see Section 5.3.1).

5.3 Alignment templates for shallow-transfer machine translation

As was introduced in Chapter 1, shallow-transfer MT systems work by parsing the SL text to translate so as to create an SL intermediate representation (IR); then, transformations are applied and the SL IR previously created is converted into a TL IR; finally, the TL text is generated from that TL IR.

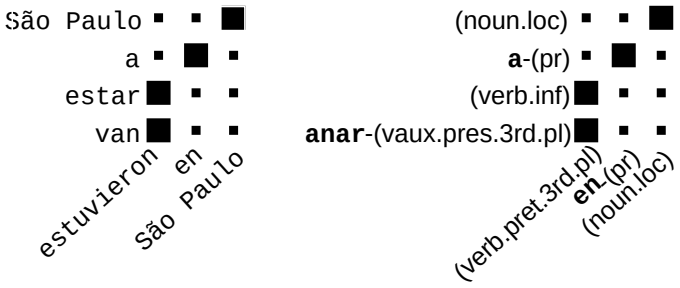
As the transformations to apply are mainly based on lexical forms, the intermediate representation used by shallow-transfer MT engines usually consists of lemma, PoS and inflection information for each word. An illustration of the intermediate representation used by the Apertium MT platform used in the experiments (Section 5.5) may be found in appendix A.

In order for the shallow-transfer MT system to benefit from the AT approach the parallel corpora must be represented in the same IR used by the translation engine. To that end, the morphological analyzers and PoS taggers of the MT system in which the transfer rules will be applied are used to analyze each side of the parallel corpus before computing the word alignments (see Section 5.2.1).

5.3.1 Word-class definition

As the transformations to apply are mainly based on the fine-grained PoS tags of SL and TL words, the function $C(\cdot)$ that maps words into word classes will map each word into a word class representing its fine-grained PoS tag, that is, a word class representing its lexical category and morphological inflection information (such as gender, number or verb tense).

Using PoS information to define the set of word classes allows the method to learn syntactic rules such as reordering and agreement rules, and verb tense changes, among others. However, in order to learn lexical changes, such as preposition changes or



$$R = \{w_1 = \texttt{verb.*}, w_3 = \texttt{noun.*}\}$$

Figure 5.3: Example of Spanish–Catalan bilingual phrases (left), AT (right) obtained when each word is replaced by its corresponding word class, and TL restrictions (see Section 5.3.2) for the Spanish-to-Catalan translation. Words in boldface correspond to lexicalized categories (see Section 5.3.1). Word classes in the horizontal axis correspond to the SL (Spanish) and in the vertical axis to the TL (Catalan).

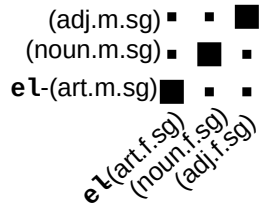
auxiliary verbs usage, some words will be assigned single word classes representing a complete lexical form, as discussed next.

Lexicalized categories

A set of (lexicalized) categories usually involved in lexical changes such as prepositions and auxiliary verbs may be provided. For those words whose lexical category is in that set of lexicalized categories (from now on, *lexicalized words*), the lemma is also used when defining the word class they belong to. In this way, lexicalized words are placed in single-word classes. For example, if prepositions are considered lexicalized categories, words *to* and *for* would be in different word classes, even if they have the same lexical category and morphological inflection information, whereas words *book* and *house* would be in the same word class (noun, singular).

Typically the set of lexicalized categories is a subset of the set of categories that do not grow by addition of new words to the lexicon: pronouns, auxiliary verbs, prepositions, conjunctions, etc. The most typical lexicalized words are prepositions, as they usually have many different translations depending on the SL context.

Figure 5.3 shows an example of a Spanish–Catalan bilingual phrase and the generalization performed when each word is replaced by its corresponding word class; words in bold face correspond to lexicalized categories. The AT shown in Figure 5.3 generalizes, on the one hand, the use of the auxiliary Catalan verb *anar* to express the past (preterite) tense and, on the other hand, the preposition change when it refers to a location name, such as the name of a city or a country.



$$R = \{w_2 = \texttt{noun.m.*}, w_3 = \texttt{adj.*}\}$$

Figure 5.4: Spanish–Catalan AT and TL restrictions over the inflection information for the Spanish-to-Catalan translation (see Section 5.3.2).

5.3.2 Extending the definition of alignment template

In Section 5.2 an AT was defined as a tuple $z = (S_m, T_n, G)$ in which only the alignment between SL and TL word classes was considered. Here the definition of AT is extended to $z = (S_m, T_n, G, R)$, where a set of restrictions, R, over the TL inflection information of non-lexicalized categories is added.

TL Restrictions

When translating (that is, when applying ATs, see next section), TL inflection information of non-lexicalized words is taken from the corresponding TL word class in the AT being applied, not from the bilingual dictionary; because of this, restrictions are needed in order to prevent an AT to be applied in certain conditions that would produce an incorrect translation. For example, an AT that changes the gender of a noun from masculine to feminine (or vice versa) would produce an incorrect TL word if such a change is not allowed for that noun. Restrictions refer to TL inflection information; therefore, they are obtained for a given translation direction and they change when translating the other way round.

TL restrictions are obtained from the bilingual dictionary of the MT system in which the inferred transfer rules will be integrated. Bilingual dictionaries may explicitly code all the inflection information of the translation of each SL lexical form, or only the inflection information that changes between the two languages. TL restrictions could be derived from both kinds of bilingual dictionaries; however, their extraction is easier if only changes in the inflection information are explicitly coded.

For the experiments (see Section 5.5) the Apertium MT platform has been used; in Apertium bilingual dictionaries, only changes in inflection information are explicitly coded. The following two examples show, on the one hand, a Spanish–Catalan bilingual

entry and, on the other hand, the restriction over the TL inflection information for the Spanish-to-Catalan translation derived for that bilingual entry:[3]

- Bilingual entry without any inflection information change

```
<e><p>
 <l>castigo<s n="noun"/></l>
 <r>càstig<s n="noun"/></r>
</p></e>
```

  Restriction: w=`noun.*`

- Bilingual entry in which the gender changes from feminine (Spanish) to masculine (Catalan)

```
<e><p>
 <l>calle<s n="noun"/><s n="f"/></l>
 <r>carrer<s n="noun"/><s n="m"/></r>
</p></e>
```

  Restriction: w=`noun.m.*`

As can be seen, restrictions provide the lexical category and morphological inflection information that the lexical form should have at translation time after looking it up in the bilingual dictionary; the star at the end of each restriction means that the rest of the inflection information is not restricted. The second bilingual entry would be responsible for the restrictions attached to w_2 in the AT shown in Figure 5.4. The AT in that figure generalizes the rule to apply in order to propagate the gender from the noun to the article and the adjective, and can only be applied if the noun (w_2) is masculine in the TL (see next section for a description of how ATs are applied).

5.4 Generation of Apertium transfer rules

This section describes the automatic generation of Apertium structural transfer rules; note, however, that the generation of transfer rules for other shallow-transfer MT systems (Canals et al., 2000; Garrido-Alenda et al., 2004) would also be feasible by following the approach presented here.

Apertium structural transfer (see appendix A) uses finite-state pattern matching to detect, in the usual left-to-right, longest-match way, fixed-length patterns of lexical

[3]Lemmas between `<l>` and `</l>` XML tags correspond to Spanish words; analogously, lemmas between `<r>` and `</r>` tags correspond to Catalan words. Inflection information is coded through the `<s>` (*symbol*) XML tag, the first one being the PoS.

forms to process and performs the corresponding transformations. A (generic) shallow-transfer rule consists of a sequence of lexical forms to detect and the transformations that have to be applied to them.

5.4.1 Filtering of the bilingual phrase pairs

In our approach not all bilingual phrase pairs can be used in the inference of transfer rules. A bilingual phrase pair may not be useful either because SL and TL non-lexicalized words are not aligned, or because it cannot be reproduced using the bilingual dictionary of the MT system. SL non-lexicalized words are required to be aligned with TL non-lexicalized words as a direct consequences of how translation is performed — by using the translation of the lemma of the SL non-lexicalized words, as provided by the bilingual dictionary, in combination with the inflection information provided by the (aligned) TL non-lexicalized words (see Section 5.4.3). The bilingual phrase pair may not be reproducible by the MT system because the translation equivalent (in the bilingual dictionary) differs from that in the bilingual phrase. In such case the set of restrictions attached to the AT could make no sense at all.

In addition, bilingual phrases are required to have their first and last words on both sides (source and target) aligned with at least one word in the other side.

5.4.2 Selecting the alignment templates to use

To decide which ATs to take into account for the generation of rules, the method is provided with a frequency count threshold. ATs whose frequency count is below this threshold are discarded. In the experiments, two different ways of interpreting the frequency count have been tested:

- to use directly the frequency count c, and
- to use a modified frequency count $c' = c(1 + \log(l))$, where l stands for the length of the SL part of the AT.

The second approach aims at solving the problem caused by the fact that longer ATs have lower frequency counts but may be more accurate as they take more context into account. A similar approach was used by Mikheev (1996) in his work on learning PoS guessing rules to prioritize longer suffixes over shorter ones.

5.4.3 Rule generation

A rule consists of a set U of ATs with the same sequence of SL word classes, but different sequences of TL word classes, different alignment information or a different

set of TL restrictions. Formally this may be expressed as follows:

$$U = \{(S_m, T_n, G, R) \in Z : S_m = S^U\}, \tag{5.2}$$

where Z refers to the whole set of ATs and S^U to the sequence of SL word classes that all ATs $z \in U$ have in common.

For each set U an Apertium shallow-transfer rule matching the sequence of SL word classes S^U is generated; that rule consists of code applying (see below) always the most frequent AT $z = (S_m, T_n, G, R) \in U$ that satisfies the TL restrictions R. A "default" AT, which translates word for word, is always added with the lowest frequency count. This AT has no TL restrictions and is the one applied when none of the remaining ATs can be applied because their TL restrictions are not met.

Code generated for each alignment template

Code is generated by following the order specified by the TL part T_n of the AT. The generated code for each unit in T_n depends on the type of its word class:

- if the word class corresponds to a non-lexicalized word, code is generated to retrieve the translation of the lemma of the aligned SL (non-lexicalized) word by looking it up in the bilingual dictionary, and to attach to the translated lemma the lexical category and morphological inflection information provided by the TL word class;
- if the word class corresponds to a lexicalized word, it is introduced as is; recall that word classes belonging to lexicalized words represent complete lexical forms consisting of lemma, lexical category and morphological inflection information.

Note that the information about SL lexicalized words is not taken into account when generating the code for a given AT. Figure 5.5 shows the code generated in the XML-based Apertium transfer language (Forcada et al., 2007, Sec. 3.5) for the AT shown in Figure 5.3.

Example of application of an alignment template

The following example illustrates how the AT shown in Figure 5.3 would be applied in order to translate from Spanish to Catalan the input text *vivieron en Francia*.[4] This text segment, after morphological analysis and PoS tagging, is transformed by the MT engine into the intermediate representation *vivir*`-(verb.pret.3rd.pl)` *en*`-(pr)` *Francia*`-(noun.loc)`, which becomes the input to the structural transfer module.

[4]Translated into English as *They lived in France*.

```
 1   <choose><when>
 2     <test><and>
 3       <or><equal><clip pos="1" side="tl" part="tags" queue="no"/>
 4                  <lit-tag v="vblex"/></equal>
 5           <equal><clip pos="1" side="tl" part="tags" queue="yes"/>
 6                  <lit-tag v="vblex.inf"/></equal></or>
 7       <or><equal><clip pos="3" side="tl" part="tags" queue="no"/>
 8                  <lit-tag v="noun"/></equal>
 9           <equal><clip pos="3" side="tl" part="tags" queue="yes"/>
10                  <lit-tag v="noun.loc"/></equal></or>
11       </and></test>
12       <out><lu><lit v="anar"/>
13                <lit-tag v="vaux.p3.pl"/>
14                <lit v=""/></lu></out>
15       <out><b pos="1"/></out>
16       <out><lu><clip pos="1" side="tl" part="lemh"/>
17                <lit-tag v="vblex.inf"/>
18                <clip pos="1" side="tl" part="lemq"/></lu></out>
19       <out><b pos="2"/></out>
20       <out><lu><lit v="a"/>
21                <lit-tag v="pr"/>
22                <lit v=""/></lu></out>
23       <out><b/></out>
24       <out><lu><clip pos="3" side="tl" part="lemh"/>
25                <lit-tag v="noun.loc"/>
26                <clip pos="3" side="tl" part="lemq"/></lu></out>
27     </when></choose>
```

Figure 5.5: Code generated in the XML-based Apertium transfer language (Forcada et al., 2007, Sec. 3.5) for the AT shown in Figure 5.3. First, TL restrictions are checked (lines 1–11) and if they hold the AT is applied (lines 12–26). Element `clip` is used to get the lemma, part-of-speech and inflection information of the SL word at the give position, or its TL translation as provided by the bilingual dictionary. Element `lit` specifies the lemma of the lexical unit (`lu`) being output (`out`); analogously, element `lit-tag` specifies the part of speech and inflection information to be attached to that lexical unit.

The AT is applied in the order specified in its TL part. For the word classes corresponding to non-lexicalized words, the aligned SL words are translated into TL (Catalan) by looking them up in the bilingual dictionary: *vivir* is translated as *viure* and *Francia* is translated as *França*. Then, the inflection information provided by the TL part of the AT (see Figure 5.3) is attached to each translated lemma. Finally, word classes corresponding to lexicalized words are just copied to the output as they appear in the TL part of the AT. For the running example the structural transfer output would be: *anar*`-(vaux.pres.3rd.pl)` *viure*`-(verb.inf)` *a*`-(pr)` *França*`-(noun.loc)`, which the generation module would transform into the Catalan phrase *van viure a França*.

5.5 Experiments

5.5.1 Task and evaluation

Task

The approach presented in this chapter was tested on both translation directions of the Spanish–Catalan (`es-ca`) and Spanish–Galician (`es-gl`) language pairs, and on the Spanish-to-Portuguese translation (`es-pt`).[5] Furthermore, two different training corpora were used for each language pair in order to test the importance of the amount of parallel corpora available for training.

As explained in Section 5.3.1, a set of categories usually involved in lexical changes needs to be provided for the definition of word classes so as to learn not only syntactic transformations, but also lexical transformations. To that end a small set with around 8 lexicalized categories is used for each language. The most common lexicalized categories are: prepositions, pronouns, determiners, subordinate conjunctions, relatives, adverbs that can precede other adverbs, modal verbs and auxiliary verbs. Similar categories have been used in example-based MT (Gough and Way, 2004; Tinsley et al., 2008) to segment the training corpus into chunks based on 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.

Regarding the length of the bilingual phrase pairs extracted and used to obtain the ATs, a maximum of 7 SL words has been established for all the experiments. Recall from Section 5.2.2 that to extract bilingual phrases from a pair of word-aligned sentences, all possible pairs (within a certain length) are considered; by restricting that length we make the problem computationally affordable.

[5] All linguistic data used can be freely downloaded from `http://sourceforge.net/projects/apertium`, packages `apertium-es-ca-1.0.2`, `apertium-es-gl-1.0.4` and `apertium-es-pt-0.9.2`.

Training corpus	# sentences	Language	# words
es-ca#1	100 834	es	1 952 317
		ca	2 032 925
es-ca#2	25 000	es	489 663
		ca	510 196
es-gl#1	89 972	es	2 073 161
		gl	1 954 177
es-gl#2	21 992	es	512 135
		gl	482 367
es-pt#1	59 972	es	1 909 304
		pt	1 836 568
es-pt#2	15 500	es	509 370
		pt	491 965

Table 5.1: Number of sentences and words in each parallel corpus used for training. The Spanish–Catalan parallel corpora come from *El Periódico de Catalunya*, the Spanish–Galician parallel corpora come from *Diario Oficial de Galicia*, and the Spanish–Portuguese parallel corpora come from *The JRC-Acquis Multilingual Parallel Corpus*.

Table 5.1 shows the number of sentences and words in the training parallel corpora; the Spanish–Catalan parallel corpora come from *El Periódico de Catalunya*,[6] a daily newspaper published both in Catalan and Spanish; the Spanish–Galician parallel corpora come from *Diario Oficial de Galicia*,[7] the official publication of the autonomous government of Galicia published both in Galician and Spanish; the Spanish–Portuguese parallel corpora come from *The JRC-Acquis Multilingual Parallel Corpus* (Steinberger et al., 2006)[8] which contains European Union (EU) law applicable in the member states of the EU.

Evaluation

The performance of the presented approach is compared to that of the same MT system when no transfer rules are used at all (word-for-word MT), and that of using hand-coded transfer rules. To that end the WER computed as the word-level *edit distance* (Levenshtein, 1965) between the translation performed by the MT system and a reference translation, and the BLEU (Papineni et al., 2002) score calculated using the same test corpus and reference translation are reported, as in the other chapters. In both cases the confidence interval of the measure being reported is given. Confidence intervals are calculated through the bootstrap resampling method (Koehn, 2004; Efron and Tibshirani, 1994) as explained in Section 2.5.1 (see page 29) .

[6] http://www.elperiodico.com
[7] http://www.xunta.es/diario-oficial
[8] http://wt.jrc.it/lt/Acquis/

Trans. dir.	Test corpus	# sentences	# SL words
es-ca	post-edit	457	10 066
	parallel	651	13 196
ca-es	post-edit	458	10 024
	parallel	651	13 737
es-gl	post-edit	382	10 398
	parallel	277	11 000
gl-es	post-edit	370	10 514
	parallel	277	10 172
es-pt	post-edit	458	10 060
	parallel	460	15 083

Table 5.2: Number of sentences and number of SL words of the two different test corpora used for the evaluation of the inferred rules for each translation direction of the Spanish–Catalan and the Spanish–Galician language pairs, and of the Spanish-to-Portuguese translation.

Table 5.2 shows the number of sentences and the number of SL words of the different test corpora used for the evaluation of the inferred rules for each translation being considered. Two different test corpora were used for each translation, one (post-edit) in which the reference translation is a post-edited (corrected) version of the MT performed when using hand-coded transfer rules, and another (parallel) in which the text to translate and the reference translation come from a parallel corpus analogous to the one used for training. It must be stressed that when evaluating using as reference translation a post-edited (corrected) version of the same MT output, BLEU scores are higher than may initially be expected, compared to the experiments reported in previous chapters. Furthermore, it is worth mentioning that the evaluations performed using the post-edited test corpora will be slightly biased towards the systems using hand-coded transfer rules, as the post-edited translations used as references are corrected versions of MT output performed using the same hand-coded transfer rules.

5.5.2 Results

Table 5.3 shows the WERs and BLEU scores, together with their respective 95% confidence intervals, achieved for each training corpus used (see Table 5.1), translation direction and evaluation test corpus (see Table 5.2) of the Spanish–Catalan language pair. The error rates reported are: (a) the results of a word-for-word translation (when no structural transformations are applied), (b) the results when the frequency count is directly used to select the set of ATs to use for the rules generation, (c) the results when a modified frequency count (see Section 5.4.2) is used to select that set of ATs, and (d) the results achieved when using hand-coded transfer rules; in all cases the same linguistic data (morphological and bilingual dictionaries) were used.

			WER (%)			
Training	**Test**	**Trans.**	**No rules**	**AT count**	**AT log**	**Hand**
es-ca#1	post-edit	es-ca	12.6 ± 0.9	8.7 ± 0.7	8.7 ± 0.7	6.7 ± 0.7
		ca-es	11.6 ± 0.8	8.1 ± 0.6	8.1 ± 0.7	6.5 ± 0.6
	parallel	es-ca	26.4 ± 1.2	20.3 ± 1.1	20.3 ± 1.1	20.7 ± 1.0
		ca-es	19.2 ± 1.0	14.7 ± 0.9	14.8 ± 0.9	14.4 ± 0.9
es-ca#2	post-edit	es-ca	12.6 ± 0.9	9.9 ± 0.7	9.9 ± 0.7	6.7 ± 0.7
		ca-es	11.6 ± 0.8	8.7 ± 0.7	8.6 ± 0.7	6.5 ± 0.6
	parallel	es-ca	26.4 ± 1.2	21.4 ± 1.1	21.3 ± 1.1	20.7 ± 1.0
		ca-es	19.2 ± 1.0	15.4 ± 0.9	15.4 ± 0.9	14.4 ± 0.9

			BLEU (%)			
Training	**Test**	**Trans.**	**No rules**	**AT count**	**AT log**	**Hand**
es-ca#1	post-edit	es-ca	65.4 ± 2.6	84.8 ± 1.2	84.6 ± 1.3	88.1 ± 1.1
		ca-es	80.0 ± 1.4	85.5 ± 1.3	85.6 ± 1.3	87.6 ± 1.0
	parallel	es-ca	46.3 ± 2.2	65.3 ± 1.6	65.2 ± 1.5	65.2 ± 1.5
		ca-es	65.0 ± 1.6	71.4 ± 1.6	71.4 ± 1.6	73.2 ± 1.4
es-ca#2	post-edit	es-ca	65.4 ± 2.6	83.1 ± 1.2	83.0 ± 1.3	88.1 ± 1.1
		ca-es	80.0 ± 1.4	84.4 ± 1.3	84.6 ± 1.3	87.6 ± 1.0
	parallel	es-ca	46.3 ± 2.2	63.5 ± 1.6	63.6 ± 1.6	65.2 ± 1.5
		ca-es	65.0 ± 1.6	70.2 ± 1.6	70.4 ± 1.6	73.2 ± 1.4

Table 5.3: WERs and BLEU scores for each training corpus, translation direction and evaluation test corpus of the Spanish–Catalan language pair. Both measures are reported with their respective 95% confidence intervals for test sets with the number of sentences reported for each test corpus in Table 5.2. The evaluation measures reported are (from left to right): the result when no transfer rules are used (No rules), the best result achieved when the count is directly used to select the set of ATs to use (AT count), the best result achieved when a modified frequency count is used to select that set of ATs (AT log, see Section 5.4.2), and the results achieved when hand-coded transfer rules are used (Hand).

As can be seen in Table 5.3 the translation quality achieved by the transfer rules inferred is better than word-for-word translation, even when a small parallel corpus (`es-ca`#2) is used for training; note that the larger training corpus (`es-ca`#1) may still be considered tiny if we compare it with the corpus sizes normally used to train SMT systems (Och, 2005). Furthermore, all confidence intervals happen to be similar regardless of the training corpus used and the transfer rules being evaluated.

Concerning the results obtained when using different test corpora, Table 5.3 shows that when evaluating via a post-edited translation, hand-coded rules perform better than automatically inferred rules; however, they give comparable results (confidence measures overlap) when using a test corpus similar to the one used for training (parallel). On the one hand, this result suggests that the automatically inferred transfer rules produce text of the same style of that used for training and that, even though they "learn" the style of the training corpus, the translation quality for other texts is quite acceptable. On the other hand, this result may be due to the fact that the post-edit evaluation is slightly biased towards the system using hand-coded transfer rules; recall that in the post-edit evaluation the reference translation is a post-edited machine translation performed with the same hand-coded transfer rules.

Table 5.4 shows the WERs and BLEU scores, together with their respective 95% confidence intervals, achieved for each training corpus used (see Table 5.1), translation direction and test corpus of the Spanish–Galician language pair. The error rates reported correspond to the same types of transfer rules that were reported for Spanish–Catalan in Table 5.3.

The Spanish–Galician language pair shows results in agreement with those obtained for Spanish–Catalan; however, the improvement on the Spanish-to-Galician translation quality, compared to word-for-word translation, is smaller. Nevertheless, the room for improvement, that is, the difference between the translation quality achieved when using hand-coded transfer rules and that of a word-for-word translation, is smaller in both translation directions of the Spanish–Galician language pair; this may indicate that Spanish and Galician are more related than Spanish and Catalan. In addition, the structural transfer rules inferred when training through the small Spanish–Galician parallel corpus (`es-gl`#2, see Table 5.1) achieve very similar results to those achieved when using the largest training corpus (`es-gl`#1); note that in the case of Spanish–Catalan the results achieved by the rules inferred using the smallest training corpus (`es-ca`#2) are worse than those attained when the rules are inferred from the biggest training corpus (`es-ca`#1), but still better than the word-for-word translation.

Concerning the Spanish-to-Portuguese translation, Table 5.5 shows the WERs and BLEU scores achieved for each training corpus and evaluation test corpus used. Both evaluation measures show that, when evaluating through the post-edit evaluation corpus, the results achieved by the AT-based inferred rules are slightly worse than those achieved by a word-for-word translation (when no transfer rules are used at all). However, when the evaluation is done using a corpus analogous to the one used for training

			WER (%)			
Training	**Test**	**Trans.**	**No rules**	**AT count**	**AT log**	**Hand**
es-gl#1	post-edit	es-gl	5.8 ± 0.6	5.5 ± 0.5	5.5 ± 0.5	3.7 ± 0.5
		gl-es	8.1 ± 0.7	7.0 ± 0.6	7.0 ± 0.6	5.8 ± 0.5
	parallel	es-gl	10.0 ± 0.7	8.3 ± 0.8	8.2 ± 0.8	8.9 ± 0.7
		gl-es	11.1 ± 0.7	10.1 ± 0.7	10.1 ± 0.7	10.2 ± 0.7
es-gl#2	post-edit	es-gl	5.8 ± 0.6	5.5 ± 0.6	5.6 ± 0.6	3.7 ± 0.5
		gl-es	8.1 ± 0.7	7.1 ± 0.6	7.1 ± 0.6	5.8 ± 0.5
	parallel	es-gl	10.0 ± 0.7	8.5 ± 0.8	8.5 ± 0.8	8.9 ± 0.7
		gl-es	11.1 ± 0.7	10.3 ± 0.7	10.3 ± 0.7	10.2 ± 0.7

			BLEU (%)			
Training	**Test**	**Trans.**	**No rules**	**AT count**	**AT log**	**Hand**
es-gl#1	post-edit	es-gl	87.3 ± 1.2	89.3 ± 1.1	89.2 ± 1.0	92.7 ± 0.9
		gl-es	70.2 ± 2.8	83.7 ± 1.9	82.7 ± 2.1	88.3 ± 1.1
	parallel	es-gl	78.3 ± 1.3	83.7 ± 1.5	83.9 ± 1.4	81.7 ± 1.3
		gl-es	64.3 ± 3.1	78.2 ± 1.9	78.0 ± 2.0	79.5 ± 1.3
es-gl#2	post-edit	es-gl	87.3 ± 1.2	89.2 ± 1.1	89.1 ± 1.1	92.7 ± 0.9
		gl-es	70.2 ± 2.8	80.4 ± 2.3	80.6 ± 2.3	88.3 ± 1.1
	parallel	es-gl	78.3 ± 1.3	83.4 ± 1.4	83.4 ± 1.5	81.7 ± 1.3
		gl-es	64.3 ± 3.1	74.2 ± 2.4	73.1 ± 2.7	79.5 ± 1.3

Table 5.4: WERs and BLEU scores for each training corpus, translation direction and test corpus of the Spanish–Galician language pair. Evaluation measures are reported with their respective 95% confidence intervals for test sets of the number of sentences reported for each test corpus in Table 5.2. The measures reported correspond to the results achieved when using different sets of transfer rules (see Table 5.3).

			WER (%)			
Training	**Test**	**Trans.**	**No rules**	**AT count**	**AT log**	**Hand**
es-pt#1	post-edit	es-pt	11.9 ± 0.8	12.1 ± 0.9	12.1 ± 0.8	7.0 ± 0.7
	parallel	es-pt	47.9 ± 1.7	46.5 ± 1.7	46.3 ± 1.7	47.6 ± 1.8
es-pt#2	post-edit	es-pt	11.9 ± 0.8	12.1 ± 0.9	12.1 ± 0.9	7.0 ± 0.7
	parallel	es-pt	47.9 ± 1.7	47.4 ± 1.7	47.4 ± 1.7	47.6 ± 1.8

			BLEU (%)			
Training	**Test**	**Trans.**	**No rules**	**AT count**	**AT log**	**Hand**
es-pt#1	post-edit	es-pt	79.1 ± 1.3	78.7 ± 1.3	78.8 ± 1.4	87.0 ± 1.2
	parallel	es-pt	35.9 ± 1.5	37.6 ± 1.5	37.9 ± 1.5	36.1 ± 1.5
es-pt#2	post-edit	es-pt	79.1 ± 1.3	78.8 ± 1.4	78.7 ± 1.4	87.0 ± 1.2
	parallel	es-pt	35.9 ± 1.5	36.4 ± 1.6	36.3 ± 1.6	36.1 ± 1.5

Table 5.5: WERs and BLEU scores for each training corpus and test corpus of the Spanish-to-Portuguese translation. Both measures are reported with their respective 95% confidence intervals for test sets with the number of sentences reported for each test corpus in Table 5.2. The measures reported correspond to the results achieved when using different sets of transfer rules (see Table 5.3).

(parallel) the translations performed using the AT-based transfer rules are better than the word-for-word translation and even better than those produced using the hand-coded transfer rules.

The evaluation performed using the parallel test corpus shows little differences between the word-for-word translation, the translation performed using hand-coded transfer rules, and that using the automatically inferred rules. This may be explained by the fact that the parallel test corpus, as well as the training corpus used to infer the Spanish-to-Portuguese structural transfer rules, has not been built by translating one language (say Spanish or Portuguese) into the other; instead, both sides of this parallel corpus are translations made from one of the European Union working languages (English, or perhaps French).

The way in which the Spanish–Portuguese parallel corpora used for training have been built may also explain the results provided by the AT-based structural transfer rules when evaluating through the post-edited test corpus. As both sides of the training corpus are translations of a third language, probably performed by different translators, the alignment information is not very reliable, as demonstrated by the percentage of discarded bilingual phrase pairs (see Section 5.4.1); this percentage is around 53 % for the Spanish-to-Portuguese translation, about 22 % for the Spanish–Catalan language pairs, and around 20 % for the Spanish–Galician language pair.

Finally, note that both criteria used to select the set of ATs to be used in the generation of transfer rules (see Section 5.4.2) give comparable results in all the experiments conducted.

5.6 Discussion

In this chapter, the generation of shallow-transfer rules from statistically-inferred ATs has been tested. To this end, a very small amount of linguistic information, in addition to the linguistic data used by the MT engine, was used in order to learn, not only syntactic changes, but also lexical changes to apply when translating SL texts into the TL. This linguistic information consists of a small set of lexical categories involved in lexical changes (prepositions, pronouns, etc.) and can be easily provided by an expert, or automatically extracted from a lexical database (Armstrong et al., 2006), if available.

The method was tested using data from three existing language pairs of the open-source shallow-transfer MT engine Apertium; more precisely, the presented approach was tested on both translation directions of the Spanish–Catalan and Spanish–Galician languages pairs, and on the Spanish-to-Portuguese translation. The performance of the system when using the automatically generated rules was compared to that of a word-for-word translation (when no structural transformations are applied) and that obtained using hand-coded transfer rules. To perform such comparison, two types of evaluation corpora were used: on the one hand, an evaluation test corpus whose reference translation is a post-edited version of the MT performed using the hand-coded transfer rules (post-edit), and, on the other hand, an evaluation test corpus analogous to the one used for training (parallel).

The evaluation of the inferred rules for both translation directions of the Spanish–Catalan and the Spanish–Galician language pairs show an improvement in the translation quality as compared to word-for-word translation for the two test corpora, but a bigger improvement is to be seen when evaluating using the parallel test corpus which provides results comparable to those achieved by using hand-coded transfer rules. In the case of the Spanish-to-Portuguese direction, there is no improvement when evaluating using the post-edited test corpus; however, the evaluation performed using the parallel test corpus shows an improvement over the word-for-word translation.

With respect to the parallel corpus used for training, the results achieved by the inferred rules for the Spanish-to-Portuguese direction show that the procedure followed to build the parallel corpus (that is, the way in which the translation from one language into the other one is performed) deserves special attention. It can be concluded that parallel corpora that have been built by translating from a third language are not appropriate for the task of inferring rules to be used in RBMT, especially if the languages involved are closely related. In contrast, the best training corpora are those built by

post-editing the output of an MT system, as is the case with the Spanish–Catalan and Spanish–Galician parallel corpora.

Finally, it is important to note that the two different criteria tested to select the set of ATs to take into account for the generation of the shallow-transfer rules —the use of a modified frequency count that tends to select longer ATs than shorter ones, and the direct use of the frequency count, which benefits shorter ATs over longer ones— give comparable results. This may be explained by the fact that, on the one hand, rules that do not apply any AT (because of TL restrictions not being met) perform a word-for-word translation. On the other hand, rules with longer ATs have more restrictions to check and, therefore, they are more likely to fail these checks and eventually perform a word-for-word translation. These results suggest that the application of shorter ATs within the same rule when none of the longer ATs can be applied could improve the results reported in this chapter.

Chapter 6

Concluding remarks

To conclude this dissertation, this chapter presents its main contributions and outlines some future research lines that may improve the results reported in the different chapters.

6.1 Summary

The main goal of all the approaches presented in this thesis has been that of easing the development of shallow-transfer RBMT systems by avoiding the need for human intervention in some of the stages of development in such MT systems. More precisely, this thesis has focused on:

- a novel, MT-oriented, unsupervised method to train the SL HMM-based PoS tagger used in an RBMT system;
- the application of a clustering strategy to automatically determine the tagset (that is, the set of states) of the HMM-based PoS tagger; and
- the automatic inference of structural transfer rules used in the shallow-transfer module of an RBMT system from a small amount of parallel corpora.

Regarding the unsupervised training of HMM-based PoS taggers to be used in MT, this thesis describes a new (MT-oriented) method that uses information from the TL and from the remaining modules of the RBMT system in which the resulting tagger is integrated; the method is evaluated on three different languages (Spanish, French and Occitan) being translated into Catalan (Chapter 2). It has been shown that MT systems that make use of HMM-based PoS taggers trained through this new MT-oriented method produce better translations than those MT systems that use PoS tagger trained using Baum-Welch (the standard "SL-only" unsupervised training

algorithm). Furthermore, the translation quality achieved is similar to that obtained when the PoS tagger is trained in a supervised way from hand-tagged corpora.

It must be noted that the comparison with a PoS tagger trained in a supervised way has only been done with one language (Spanish), as hand-tagged texts were not available for the other two languages, and that the hand-tagged corpus used for the supervised training was no so large. In any case, the lack of larger hand-tagged corpora, or the absence of hand-tagged corpora at all, motivates the need for a training method like the one proposed in this thesis which makes use of a source of knowledge that is readily available when developing PoS taggers to be used in MT: a statistical language model of the TL.

To benefit from this new method when building an RBMT system from scratch, system developers only need to build the rest of the modules of the translation engine before training the HMM-based PoS taggers of that MT system. This implies developing monolingual and bilingual dictionaries and structural transfer rules. However, when the RBMT system being developed involves closely-related languages, the PoS taggers can be trained before having a complete set of structural transfer rules, since the results achieved when no structural transfer rules are used during training are quite similar to those achieved when using a complete structural transfer module.

This is the first time that an algorithm to train a PoS tagger for a particular language makes use of information from another language without using any parallel corpora. Furthermore, this is the first training algorithm that tries to maximize the quality of the whole translation task, instead of focusing only on PoS tagging accuracy. The initial hypothesis, that a PoS tagger that is accurate from the MT viewpoint may be less accurate as a PoS tagger in its own, has been validated. The experiments conducted on the Spanish–Catalan language pair show that a PoS tagger trained in a supervised way performs better than a PoS tagger trained with the MT-oriented algorithm if both taggers are evaluated in isolation (PoS tagging performance evaluation), but that their performances are quite similar when the PoS taggers are evaluated through the translation quality achieved by the MT systems embedding them.

The new PoS tagging training method works by translating into the TL each possible disambiguation of SL segments. As a consequence of that, the MT-oriented training method needs to perform a huge number of translations. To solve this problem, a method that uses *a priori* knowledge, also obtained in an unsupervised way, to avoid translating unlikely disambiguations has been proposed in Chapter 3. This method is based on the following hypothesis; a model of SL tags that is not accurate enough to disambiguate an SL segment may be accurate enough to select a subset of disambiguations between all of the possible ones, the correct one being included in that subset. This approach has been tested on the same three language pairs and in all cases the number of translations to perform can be drastically reduced without affecting the performance of the MT system embedding the resulting PoS tagger.

The MT-oriented method of training PoS taggers avoids the need for hand-tagged corpora to achieve better results than the Baum-Welch training algorithm; however, it does not avoid the need to manually define the HMM topology, that is, the set of hidden states to be used for the PoS tagging. In Chapter 4 the application of a clustering algorithm to reduce the number of states was tested. The clustering algorithm was applied over the set of states of an initial HMM that has as many states as different fine-grained PoS tags that the morphological analyzer delivers; this initial model was trained via the MT-oriented training method previously discussed in Chapter 2. The approach was tested on the same three language pairs; for one of them, the translation quality achieved by the MT system embedding the resulting PoS tagger was not affected. The other two language pairs show that translation quality decreases as the number of states is reduced through the proposed clustering algorithm. This reduction in quality is not very significant and can be assumed if a reduction in the number of states is crucial for the PoS taggers being used in RBMT; for example, because translation speed is an important issue, as a reduced number of states implies a reduced number of parameters to retrieve before tagging the SL text to translate.

This thesis also focused on the development of structural transfer rules to be used in MT, and more precisely on the inference of shallow-transfer rules. Chapter 5 describes how to extend the AT approach introduced in the SMT framework in order to use it to generate shallow-transfer rules to be used in RBMT. The approach has been tested on both translation directions of two different language pairs and on one translation direction of a third language pair; for each language pair two different training corpora were used: one with around two million words and another one with only half a million words. The experimental results show that the translation quality improves, compared to applying no rules at all, and that it is close to that achieved when translating using hand-coded transfer rules.

This is the first approach that extends the AT approach for use in RBMT; an important property of the inferred rules is that they can be edited by human beings so as to improve them. This means that developers of RBMT systems can use this method to obtain a set of initial transfer rules that can be then refined by linguists; proceeding in this way, human experts can focus on the more difficult issues of writing accurate transfer rules for MT, as most of the needed rules are automatically obtained from parallel corpora. From my point of view, this is a great advantage over other corpus-based approaches to MT, such as SMT, because in this approach automatically generated rules can coexist with hand-coded ones.

Finally, it must be mentioned that all the methods described in this thesis have been released as open-source software under the GNU GPL license (see appendix C). The public availability of the source code ensures the reproducibility of all the experiments conducted and allows other researchers to improve the algorithms discussed here, saving them from having to implement all the algorithms once again. In addition, all the methods have been implemented in such a way that they integrate with the Apertium open-source MT platform (see appendix A); this benefits, on the one hand, other

researchers that use Apertium as a research platform, and on the other hand, people developing new language pairs for Apertium.

6.2 Future research lines

What follows is a list of open research lines that may be followed to study more in depth some of the approaches proposed in this thesis:

1. The use of TL information has been applied in the training of HMM-based PoS taggers (see Chapter 2); however, this approach could also be used to train PoS taggers for RBMT not based on HMM, such as PoS taggers based on maximum entropy models (Ratnaparkhi, 1996, 1998) or sliding-window PoS taggers (Sánchez-Villamil et al., 2004).

2. The application of TL information to train a statistical model that runs on the SL may also be applied to train other models that may be used in RBMT; for instance, TL information could be used to tackle at an early (SL) stage the problem of *lexical selection*, that is, the problem of choosing the correct translation for those words that, according to the bilingual dictionary, can be translated in more than one way because of polysemy.

3. Another possible research line is *triangulation*, that is, the use of more than one TL to train an MT-oriented SL PoS tagger; for instance, a Spanish PoS tagger could be trained by using information from both Catalan and Portuguese by using data from the translation into these two languages. This approach may reduce the incidence of the free-ride phenomenon (SL words being translated into the same TL word for every possible disambiguation), as it may happen when translating to one of the languages but not when translating to the other one. However, this triangulation presents a number of open issues, such as the homogenization of the SL dictionaries used by each MT system, that would need to be solved in advance.

4. As a consequence of the free-ride phenomenon, more than one disambiguation path may produce the same translation. In all the experiments conducted along this thesis the fractional contribution of disambiguation path $\boldsymbol{g}$ to the translation into TL $\tau(\boldsymbol{g}, s)$ of SL segment s has been approximated as being equal. A possible research line could focus on better estimates to that fractional contribution. On the one hand, an initial model like the one used to prune unlikely disambiguation paths may be used to better estimate that factor; on the other hand, the expectation-maximization algorithm may be applied iteratively to better estimate that contribution, but at the cost of increasing the overall training time.

5. Concerning the language model used to train the PoS taggers for use in RBMT, in all the experiments a TL model based on trigrams of words has been used. It would be an interesting idea to test n-grams models of higher order, or other types of language models such as bag-of-words models.

6. In Chapter 2 we saw that the MT-oriented PoS tagging training method can be applied by using a context-free word-for-word transfer module between related languages without a significant loss in accuracy. These results suggest a new research line: whether a dynamic programming algorithm may be devised to reduce the time complexity of null-transfer training when MT involves closely-related languages. This could be done because when a null structural transfer MT system is used each word is processed after the PoS tagger independently from the adjacent ones, allowing the translation model to be described by an analytical function.

7. As for the path pruning method described in Chapter 3, additional strategies could be tested to select the set of disambiguation paths to take into account. One possible approach is the use of a method that dynamically changes the probability mass threshold during training (an *annealing schedule*; Kirkpatrick et al. 1983); this method could start with a large probability mass threshold, close to 1.0, which could be reduced as the training method proceeds.

8. Another possibility for the pruning of disambiguation paths is the use of a fixed number of disambiguation paths to translate per segment (k-best) instead of using a probability mass threshold. This method could be implemented in such a way in which all *a priori* likelihoods do not need to be explicitly calculated (Schwartz and Chow, 1990; Purnhagen, 1994) before discarding many of them.

9. The bottom-up agglomerative clustering described in Chapter 4 to automatically infer the set of states of the HMM uses the *intrinsic discrepancy* to measure the distance between two fine-grained PoS tags. A possible research line may focus on the effect of other distance measures, such as the Jensen-Shannon divergence (Grosse et al., 2002), on the clusters inferred.

10. To measure the distance between two clusters (which may contain more than one fine-grained PoS tag), the *pair-group average* was applied in Chapter 4. Another possible research line may study other measures such as the *minimum pair-group distance*, which is expected to produce clusters with more dispersed elements and the *maximum pair-group distance* which usually gives more compacted clusters (Duda et al., 2001, ch. 10).

11. With respect to the automatic inference of shallow-transfer rules from parallel corpora, in Chapter 5 two different criteria have been tested to select the set of ATs finally taken into account for the generation of transfer rules. Both criteria give comparable results, however the criterion that uses a modified frequency count that prioritizes longer ATs over shorter ones produces rules that are more

likely to eventually finish by performing a word-for-word translation. The application on shorter ATs within the same rule when none of the longer ones can be applied may improve the results achieved (Caseli et al., 2006), as this gradual "back-off" would avoid falling back straight into word-for-word translation as it is currently implemented.

12. The set of bilingual phrase pairs that are generalized to ATs are extracted from the bilingual parallel corpus without following linguistic criteria, that is, bilingual phrase pairs are extracted without worrying whether they are syntactic constituents or not. A possible research line that may improve the results could focus on a smarter, linguistically-driven extraction of the bilingual phrase pairs; for instance, by segmenting the training corpora using marker-based chunkers (Gough and Way, 2004) and then by aligning the chunks through a chunk-based alignment algorithm (Tinsley et al., 2008, sec. 2.2).

13. Another research line that may improve the results would focus on a more flexible way to use lexicalized categories. It would be of interest to have context-dependent lexicalized categories, that is, categories which are lexicalized only in some contexts, while not in others; this would improve the generalization performed by the ATs.

14. Finally, possible future work may focus on a bootstrapping approach to build both the PoS taggers and the structural transfer modules together. Note that, on the one hand, the MT-oriented method to train the PoS taggers makes use of the structural transfer to translate into the TL all possible disambiguations of SL segments, and that, on the other hand, to infer rules from a parallel corpus both sides of that corpus need to be analyzed and tagged.

Appendix A

Apertium: an open-source shallow-transfer machine translation platform

This appendix briefly describes the open-source shallow-transfer MT engine Apertium used along this thesis as a research platform. Apertium may be used to build MT systems for a variety of language pairs; to that end, the platform uses simple standard formats to encode the linguistic data needed and documented procedures to build those data and to train the necessary modules. This appendix briefly describes the machine translation engine, the formats it uses for linguistic data, and the compilers that convert these data into an efficient format used by the engine.[1]

A.1 Introduction

The open-source MT platform Apertium (Corbí-Bellot et al., 2005; Armentano-Oller et al., 2006) uses finite-state transducers for lexical processing, hidden Markov models for part-of-speech (PoS) tagging, and finite-state pattern-matching for structural transfer; the initial design was largely based upon that of systems already developed by the Transducens group[2] at the Universitat d'Alacant[3] such as interNOSTRUM[4] (Spanish–Catalan, Canals-Marote et al. 2001) and Traductor Universia[5] (Spanish–Portuguese, Garrido-Alenda et al. 2004).

[1] This appendix is largely based on a paper by Armentano-Oller et al. (2006) that describes the Apertium MT platform.

[2] `http://transducens.dlsi.ua.es`

[3] `http://www.dlsi.ua.es`

[4] `http://www.internostrum.com`

[5] `http://traductor.universia.net`

The Apertium MT platform consists of two basic packages: `lttoolbox` (containing all the lexical processing modules and tools) and `apertium` itself (containing the rest of the engine and tools). The platform is released under an open-source license (GNU GPL[6]), and includes additional packages, some of them developed as a part of this thesis (see appendix C). In addition to the MT platform, open-source data are available for a number of language pairs, such as the language pairs being used in this thesis: Spanish–Catalan (`apertium-es-ca`), Occitan–Catalan (`apertium-oc-ca`), French–Catalan (`apertium-fr-ca`), Spanish-Portuguese (`apertium-es-pt`), and Spanish–Galician (`apertium-es-gl`). All these packages can be freely downloaded from `http://sf.net/projects/apertium`.

The following sections give an overview of the architecture (Section A.2), the formats defined to encode the linguistic data (Section A.3), and the compilers used to convert these data into an executable form (Section A.4). For a complete description of Apertium we refer the reader to the documentation of the RBMT platform (Forcada et al., 2007).

A.2 The Apertium MT architecture

Apertium is a classical shallow-transfer or transformer system consisting of an 8-module assembly line; the strategy used in Apertium is sufficient to achieve a reasonable translation quality between closely related languages such as Spanish and Portuguese.[7]

To ease diagnosis and independent testing, modules communicate between themselves using text streams (examples below give an idea of the communication format used). This allows for some of the modules to be used in isolation, independently from the rest of the MT system, for other natural-language processing tasks, or for research purposes. The modules are organized as in the diagram shown in Figure A.1.

Most of the modules are capable of processing tens of thousands of words per second on current desktop workstations; only the structural transfer module lags behind at several thousands of words per second. The following sections describe each module of the shallow-transfer architecture in detail.

A.2.1 De-formatter

The de-formatter separates the text to be translated from the format information (RTF, HTML, etc.). Format information is encapsulated in square brackets so that the rest

[6] `http://www.gnu.org/licenses/gpl.html`

[7] Recently Apertium has been enhanced to deal with less-related language pairs (such as English and a Romance language); this enhancement, which only affects structural transfer and leads to 10-modules assembly line, is not discussed here as it is not used on any of the experiments conducted.

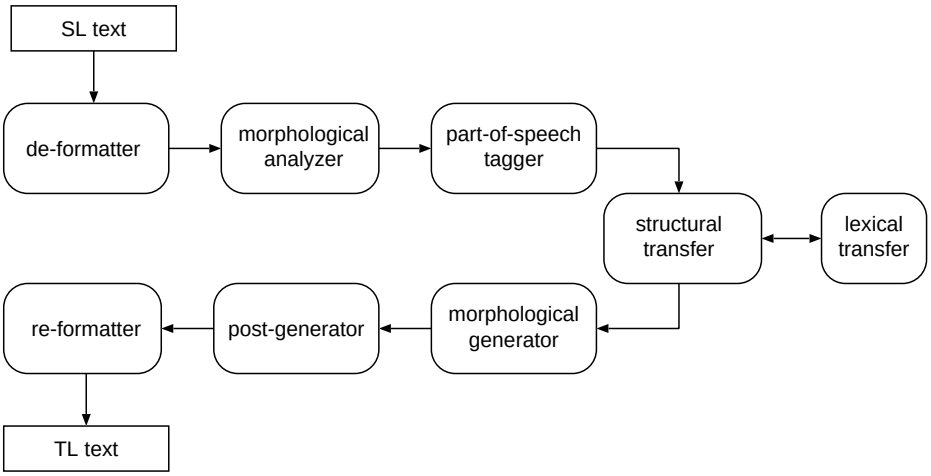


Figure A.1: Modules of the Apertium shallow-transfer MT platform (see Section A.2).

of the modules treat it as simple blanks between words. For example, the HTML text in Spanish:

```
vi <em>una señal</em>
```

("I saw a signal") would be processed by the de-formatter so that it would encapsulate the HTML tags between brackets and deliver

```
vi[ <em>]una señal[</em>]
```

As usual, the escape symbol \ is used before symbols `[` and `]` if present in the text.

A.2.2 Morphological analyzer

The morphological analyzer segments the text in *surface forms* (lexical units as they appear in texts) and delivers, for each surface form, all its possible *lexical forms* consisting of lemma, lexical category and morphological inflection information. Tokenization is not straightforward due to the existence, on the one hand, of contractions, and, on the other hand, of multi-word lexical units. For contractions, the system reads in a single surface form and delivers the corresponding sequence of lexical forms (for instance, the Spanish preposition-article contraction *del* would be analyzed into two lexical forms, one for the preposition *de* and another one for the article *el*). Multi-word surface forms are analyzed in a left-to-right, longest-match fashion; for instance, the analysis for the Spanish preposition *a* would not be delivered when the input text is *a través de* ("through"), which is a multi-word preposition in Spanish. Multi-word surface forms may be invariable (such as a multi-word preposition or conjunction) or inflected (for example, in Spanish, *echaban de menos*, "they missed", is a form of the

imperfect indicative tense of the verb *echar de menos*, "to miss"). Apertium offers support for most types of inflected multi-word units. The morphological analysis module reads in a binary file compiled from a source-language morphological dictionary (see Section A.3.1).

Upon receiving the example text in the previous section, the morphological analyzer would deliver

```
^vi/ver<vblex><ifi><1><sg>$[ <em>]
^una/un<det><ind><f><sg>/unir<vblex><prs><1><sg>/
unir<vblex><prs><3><sg>$ ^señal/señal<n><f><sg>$[</em>]
```

where each surface form is analyzed into one or more lexical forms. For example, *vi* is analyzed into lemma *ver*, lexical category verb and inflection information indefinite indicative, 1st person, singular, whereas *una* receives three analyses: *un*, determiner, indefinite, feminine singular, and two forms of the present subjunctive of the verb *unir* ("to join"). The characters "^" and "$" delimit the analyses for each surface form; lexical forms for each surface form are separated by "/"; angle brackets "<...>" are used to delimit grammatical symbols. The string after the "^" and before the first "/" is the surface form as it appears in the source input text.

A.2.3 Part-of-speech tagger

For those words with more than one lexical form (PoS tag), the PoS tagger chooses one of them, according to the lexical forms of neighboring words. The PoS tagger is based on hidden Markov models (HMM: Cutting et al. 1992; see appendix B). The HMM is trained from corpora and a tagger definition file (see Section A.3.2) that specifies how the fine-grained PoS tags delivered by the morphological analyzer must be grouped into coarse tags; for further information on the tagset definition read Section 1.4.2 in the introductory chapter.

The result of processing the example text delivered by the morphological analyzer with the PoS tagger would be

```
^ver<vblex><ifi><1><sg>$[ <em>]^un<det><ind><f><sg>$
^señal<n><f><sg>$[</em>]
```

where the correct lexical form (determiner) has been selected for the word *una*.

A.2.4 Lexical transfer

The lexical transfer module is called by the structural transfer module (see next section); it reads each SL lexical form and delivers a corresponding TL lexical form. The

module reads in a binary file compiled from a bilingual dictionary (see Section A.3.1). The dictionary contains a single equivalent for each SL entry; for some words, multi-word entries are used to safely select the correct equivalent in frequently-occurring fixed contexts.

Each of the lexical forms in the running example would be translated into Catalan as follows:

```
ver<vblex> ⟶ veure<vblex>
un<det> ⟶ un<det>
señal<n><f> ⟶ senyal<n><m>
```

where the remaining grammatical symbols for each lexical form would be simply copied to the TL output. Note the gender change from feminine (<f>) to masculine (<m>) when translating *señal* into Catalan *senyal*.

A.2.5 Structural transfer

The structural transfer module interprets a slightly preprocessed version of a structural transfer specification file (see Section A.3.3); it uses finite-state pattern matching to detect (in the usual left-to-right, longest-match way) fixed-length patterns of lexical forms (chunks or phrases) needing special processing due to grammatical divergences between the two languages (gender and number changes to ensure agreement in the TL, word reorderings, lexical changes such as changes in prepositions, etc.) and performs the corresponding transformations.

In the running example, a determiner-noun rule is used to change the gender of the determiner so that it agrees with the noun, and another rule introduces the auxiliary Catalan verb "anar" and changes the tense of the verb Catalan "veure" to infinitive; the result is

```
^anar<vbaux><pres><1><pl> ^veure<vblex><inf>$[ <em>]
^un<det><ind><m><sg> ^senyal<n><m><sg>$[</em>]
```

A.2.6 Morphological generator

The morphological generator delivers a TL surface form for each TL lexical form, by suitably inflecting it. The module reads in a binary file compiled from a TL morphological dictionary (see Section A.3.1). The result for the running example would be

```
vaig veure[ <em>]un senyal[</em>]
```

A.2.7 Post-generator

The post-generator performs orthographic operations such as contractions and introduction of apostrophes. The module reads in a binary file compiled from a rule file expressed as a dictionary (Section A.3.1). The post-generator is usually *dormant* (just copies the input to the output) until a special alarm symbol contained in some TL surface forms *wakes it up* to perform a particular string transformation if necessary; then it goes *back to sleep*.

For example, in Catalan, proclitic pronouns in contact may change: *em* ("to me") and *ho* ("it") contract into *m'ho*, *em* and *els* ("them") contract into *me'ls* and *em* and *la* ("her") are written *me la*. To signal these changes, linguists prepend an "alarm" symbol to the TL surface form "em" in TL dictionaries and write post-generation rules to effect the changes described.

A.2.8 Re-formatter

Finally, the re-formatter restores the format information encapsulated by the deformatter into the translated text and removes the encapsulation sequences used to protect certain characters in the source text. The result for the running example would be the correct translation of the HTML text:

```
vaig veure <em>un senyal</em>
```

A.3 Formats for linguistic data

The formats used by this architecture are declarative and based on XML[8] for interoperability; in particular, for easier parsing, transformation, and maintenance. Moreover, the use of well-defined XML formats allow third-party tools to automatically generate data, such as bilingual dictionaries or transfer rules, to be used by the translation engine. The XML formats for each type of linguistic data are defined through conveniently-designed XML document-type definitions (DTDs) which may be found inside the `apertium` package.

A.3.1 Dictionaries (lexical processing)

Monolingual morphological dictionaries, bilingual dictionaries and post-generation dictionaries use a common format, defined by DTD `dix.dtd`.

[8] `http://www.w3.org/XML/`

Morphological dictionaries establish the correspondences between surface forms and lexical forms and contain: (a) a definition of the alphabet (used by the tokenizer), (b) a section defining the grammatical symbols used in a particular application to specify lexical forms (symbols representing concepts such as *noun*, *verb*, *plural*, *present*, *feminine*, etc.), (c) a section defining paradigms (describing reusable groups of correspondences between parts of surface forms and parts of lexical forms[9]), and (d) one or more labeled dictionary sections containing lists of surface form–lexical form correspondences for whole lexical units (including contiguous multi-word units). Paradigms may be used directly in the dictionary sections or be nested to build larger paradigms.

Bilingual dictionaries have a very similar structure and establish correspondences between SL lexical forms and TL lexical forms, but seldom use paradigms.

Finally, post-generation dictionaries are used to establish correspondences between input and output strings corresponding to the orthographical transformations to be performed by the post-generator on the TL surface forms generated by the generator.

A.3.2 Tagset definition

SL lexical forms delivered by the morphological analyzer are defined in terms of fine-grained PoS tags (for example, Spanish word *cantábamos* ("we sang") has lemma *cantar* ("sing"), lexical category verb, and inflection information: indicative, imperfect, 1st person, plural), which are necessary in some parts of the MT engine (structural transfer, morphological generation); however, for the purpose of efficient disambiguation, these fine-grained PoS tags may be grouped in coarser tags (such as verb in personal form).

The tagger definition file is also an XML file (with the corresponding DTD file, `tagger.dtd`) where (a) coarser tags are defined in terms of fine-grained tags, both for single-word and for multi-word units, (b) constraints may be defined to forbid or enforce certain sequences of PoS tags,[10] and (c) priority lists are used to decide which fine-grained PoS tag to pass on to the structural transfer module when the coarse PoS tag contains more than a fine-grained tag.

A.3.3 Structural transfer rules

An XML format for shallow structural transfer rules has also been established; a commented DTD (`transfer.dtd`) may be found inside the `apertium` package.

Structural transfer rule files contain pattern–action rules which describe what has to be done for each pattern (much like in languages such as `perl` or `lex`). Patterns

[9] At the conceptual level, paradigms represent the regularities in the inflective system of the corresponding language.

[10] These constraints have not been used in the experiments regarding the PoS tagger in Chapters 2, 3 and 4.

are defined in terms of categories which are in turn defined (in the preamble) in terms of fine-grained morphological tags and, optionally, lemmas for lexicalized rules. For example, a commonly used pattern, determiner-noun, has an associated action which sets the gender and number of the determiner to those of the noun to ensure gender and number agreement.

A.4 Compilers and preprocessors

Apertium contains compilers to convert the linguistic data into the corresponding efficient (binary) form used by the modules of the engine. Two main compilers are used: one for the four lexical processing modules of the system and another one for the structural transfer.

A.4.1 Lexical processing

The (stand-alone) lexical processor compiler uses advanced transducer building strategies and the minimization of partial finite-state transducers (Ortiz-Rojas et al., 2005; Roche and Schabes, 1997) during construction. This makes the compilation on the linguistic data very fast, easing the development of linguistic data.

The four lexical processing modules (morphological analyzer, lexical transfer, morphological generator, post-generator) read binary files containing a compact and efficient representation of a class of finite-state transducers (in particular, augmented letter transducers, as in Garrido-Alenda et al. 2002).

A.4.2 Structural transfer

The structural transfer preprocessor reads in a structural transfer rule file and generates a file with pre-compiled patterns and indexed versions of the actions of the rules of the structural transfer module specification, ready to be used by the corresponding module (Section A.2.5).

Appendix B

Hidden Markov models for part-of-speech tagging

This appendix describes in detail the use of hidden Markov models as part-of-speech taggers in the field of natural language processing.[1] The two classical training methods are reviewed: on the one hand, the unsupervised Baum-Welch expectation-maximization (EM) algorithm, and on the other hand, the supervised maximum likelihood estimate (MLE) method. Finally, the Viterbi algorithm used to disambiguate a given input text is also reviewed.

A hidden Markov model (HMM) (Rabiner, 1989; Baum and Petrie, 1966) is a statistical model in which the system being modeled is assumed to be a Markov process with hidden states. HMMs are used for a wide variety of applications such as speech recognition, optical character recognition, statistical machine translation and part-of-speech tagging, among others.

In a regular Markov model, states are visible, and the transition probabilities between them are the only parameters to learn. In contrast, in an HMM, states are not directly visible; only observable outputs generated by the states are visible. Each state has a probability distribution over the possible observable outputs; therefore, the sequence of observable outputs generated by an HMM gives some information about the underlying sequence of hidden states.

Formally, an HMM (Rabiner, 1989; Baum and Petrie, 1966) is defined as $\lambda = (S, V, A, B, \pi)$, where $S = \{s_1, s_2, \ldots, s_N\}$ is the set of hidden states, $V = \{v_1, v_2, \ldots, v_M\}$ is the set of observable outputs, $A = \{a_{ij} : i, j = 1, \ldots, N\}$ are the state to state transition probabilities, $B = \{b_j(k) : j = 1, \ldots, N,\ k = 1, \ldots, M\}$ are the probabilities of each observable output v_k being emitted from each hidden state s_j, and

[1] This appendix is based on an appendix about the use of HMM for PoS tagging by Pérez-Ortiz (2002) which in turn contains material from a personal communication by Drs. Rafael C. Carrasco and Mikel L. Forcada.

$\pi = \{\pi_i : i = 1, \ldots, N\}$ defines the probability of each hidden state s_i being the initial one.[2] The system produces an output each time a state is reached after a transition.

B.1 Part-of-speech tagging with HMMs

When an HMM is used to perform PoS tagging, each HMM state s_i is made to correspond to a different part-of-speech (PoS) tag, and the observable outputs V are made to correspond to *word classes*. Typically a word class is an *ambiguity class* (Cutting et al., 1992), that is, the set of all possible PoS tags that a word could receive. However, sometimes it may be useful to have finer classes, such as a word class containing only a single, very frequent, ambiguous word. In addition, unknown words, that is, words not found in the lexicon, are usually assigned the ambiguity class consisting of the set of *open* categories, that is, the set of PoS tags (categories) which are likely to grow by addition of new words to the lexicon: nouns, verbs, adjectives, adverbs and proper nouns.

The PoS ambiguity is solved by assigning to each word the PoS tag found in the PoS tag sequence that maximizes its likelihood given the observable outputs. The model assumes that the PoS tag of each word depends only on the PoS tag of the previous word when a first-order HMM is used, or on that of the n preceding words when a n-th order HMM is considered.

B.1.1 Assumptions

For the purpose of PoS tagging we can make the following assumptions:

1. The text sequence $O_1 \ldots O_T$ being disambiguated is always preceded by an unambiguous word $O_0 = \{I\}$. A reasonable choice for I is the PoS tag representing the end-of-sentence mark; this makes the PoS tagging of each sentence independent of the position of the text in which they appear.

2. The input text is ended by an unambiguous word $O_T = \{E\}$; in this case the tag representing the end-of-sentence mark is also a reasonable choice for E because well-written texts are usually ended by the end-of-sentence mark.

3. All word classes (observable outputs) contain, at least, the correct PoS tag. For example, from the state associated to the PoS tag `noun` it is impossible to generate the ambiguity class {`adjective`, `verb`}; consequently, an ambiguity class $\{X\}$

[2] Along this thesis a different notation has been used to define an HMM; changes in the notation are justified by the fact that, in following sections, small modifications introduced in the formulas by Rabiner (1989) are discussed using the notation in that paper.

holding only one state (or PoS tag) can only be emitted from the corresponding state X.

B.2 Parameter estimation

The process of estimating the HMM parameters consists in finding the set of parameters that maximizes the mathematical expectation of the observed sequences. The classic methods to estimate these parameters are:

- training the model in an unsupervised way with untagged corpora via the Baum-Welch expectation-maximization algorithm (Baum, 1972; Manning and Schütze, 1999, p. 333), as described in Section B.3, or
- training the model in a supervised manner with hand-tagged corpora via the maximum-likelihood estimate (MLE) (Gale and Church, 1990), as described in Section B.4.

B.2.1 Parameter smoothing

Independently of the method used to estimate the HMM parameters, a smoothing technique should be used in order to avoid null probabilities for those state-to-state transitions and output emissions that have not been seen in the training corpus.

Parameter smoothing can be conveniently achieved using a form of *deleted interpolation* (Jelinek, 1997, ch. 4) in which weighted estimates are taken from first-order models and a uniform probability distribution, as in the TL model described in Section 2.4 (page 24).[3]

State-to-state transition probabilities

The smoothing of the state-to-state transition probabilities in A consists of a linear combination of bigram and unigram probabilities:

$$\overline{a_{ij}} = \mu(s_i)\frac{n(s_i s_j)}{n(s_i)} + (1 - \mu(s_i))P(s_j), \tag{B.1}$$

where $\mu(s_i)$ is the smoothing coefficient for tag bigrams, $n(s_i s_j)$ is the count of the number of times state s_i is followed by state s_j in the training corpus, $n(s_i)$ is the number of occurrences of state s_i in the training corpus, and $P(s_j)$ is the probability of having seen the tag s_j.

[3]The equations provided here can be easily extended to smooth the parameters of a higher-order HMM.

As for the TL model (see Section 2.4), the smoothing coefficients are estimated via the *successive linear abstraction* approximation (Brants and Samuelsson, 1995):

$$\mu(s_i) = \frac{\sqrt{n(s_i)}}{1 + \sqrt{n(s_i)}}. \tag{B.2}$$

Nevertheless, in spite of the smoothing techniques used, when a tag bigram ends at a previously unseen tag s_j, the final probability is still zero because the unigram probability $P(j)$ is null. To avoid this problem, unigram probabilities are smoothed as well via the following equation:

$$P(s_j) = \eta \frac{n(s_j)}{\sum_{s_k \in S} n(s_k)} + (1 - \eta)\frac{1}{N}, \tag{B.3}$$

in which the second term estimates, in absence of further information, the probability of each tag as being equally likely.[4] The weight of this second term in the final smoothed probability $P(j)$ depends on the smoothing coefficient η, which is made to depend on the length L of the training corpus, and may be approximated in an analogous way to that proposed by Brants and Samuelsson (1995):

$$\eta = \frac{\sqrt{L}}{1 + \sqrt{L}}. \tag{B.4}$$

Emission probabilities

Even if observable outputs are made to correspond to word classes (see Section B.1) to reduce the total number of observable outputs and the data sparseness problem, emission probabilities still need to be smoothed.

The smoothing of the emission probabilities (B) is done in an analogous way to that of the state-to-state transition probabilities:

$$\overline{b_j}(k) = \mu(s_j)\frac{n(v_k, s_j)}{n(s_j)} + (1 - \mu(s_j))P_{s_j}(v_k) \tag{B.5}$$

where $\mu(s_j)$ is the smoothing coefficient calculated as shown in Equation (B.2), $n(v_k, s_j)$ is the count of the number of times word class v_k is emitted from tag s_j, and $P_{s_j}(v_k)$ is the probability of word class v_k taking into account only those word classes which can be effectively emitted from tag s_j:

$$P_{s_j}(v_k) = \begin{cases} \frac{P(v_k)}{\sum_{v_l : s_j \in v_l} P(v_l)} & \text{if } s_j \in v_k \\ 0 & \text{otherwise} \end{cases} \tag{B.6}$$

[4] This can be easily done because the total number of tags is known and equal to N.

where $P(v_k)$ is the (smoothed) probability of ambiguity class v_k. This probability is smoothed in an analogous way to that used for $P(s_j)$ (see Equation B.3):

$$P(v_k) = \eta \frac{m(v_k)}{\sum_{v_l \in V} m(v_l)} + (1 - \eta)\frac{1}{M}, \tag{B.7}$$

where η refers to the smoothing coefficient calculated as shown in Equation (B.4), and $m(v_k)$ is the count of the number of times ambiguity class v_k is shown in the training corpus. As in Equation (B.3), in the absence of further information, all word classes are assumed to be equally likely (second term).

Equation (B.5) does not directly use the probability $P(v_k)$, because the use of $P(v_k)$ would cause the probability $b_j(k)$ to be non-null also in those cases in which $s_j \notin v_k$, that is, in those cases in which v_k cannot be emitted from tag s_j.

B.3 Baum-Welch expectation-maximization algorithm

The Baum-Welch algorithm is a special case of the *expectation-maximization* (EM) method. This training algorithm works as follows: As the model is unknown, the probability of the observation sequence can be worked out with an initial model that may be randomly chosen, or estimated from corpora via Kupiec's method (see Section B.3.6) or any other reasonable initialization method. Once an initial model is chosen the method works by giving the highest probability to the state transitions and output emissions used the most. In this way a revised model that is more accurate is obtained. This model can in turn be reestimated using the same procedure iteratively. After each Baum-Welch iteration the new HMM parameters may be shown to give a higher probability to the observed sequence (Baum, 1972). Now follows a brief mathematical justification of the Baum-Welch algorithm using the notation by Rabiner (1989).

B.3.1 Forward probabilities

As the text sequence being analyzed has been assumed to be preceded by an unambiguous word $O_0 = \{I\}$, the estimation of the initial probability of each state is not necessary because its value can be fixed beforehand; $\pi_i = \delta_{I,s_i}$, where δ is the Kronecker delta defined as:

$$\delta_{i,j} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{otherwise} \end{cases} \tag{B.8}$$

Given that $\pi_i = \delta_{I,s_i}$ and that $b_i(\{I\}) = \delta_{I,s_i}$ Equations (19) and (20) by Rabiner (1989) can be rewritten beginning at $t = 0$ as:

$$\alpha_0(i) = \delta_{I,s_i} \tag{B.9}$$

and, for $t = 1, \ldots, T$ as

$$\alpha_t(i) = \sum_{j=1}^{N} \alpha_{t-1}(j)\ a_{ji}\ b_i(O_t). \tag{B.10}$$

B.3.2 Backward probabilities

Analogously, backward variables, corresponding to Equations (24) and (25) by Rabiner (1989), are:

$$\beta_T(i) = 1 \tag{B.11}$$

and, for $t = 1, \ldots, T$,

$$\beta_{t-1}(i) = \sum_{j=1}^{N} a_{ij}\ b_j(O_t)\ \beta_t(j) \tag{B.12}$$

B.3.3 Other probabilities

The probability of a sequence $\boldsymbol{O} = O_1 \ldots O_T$ can be worked out using the forward and backward probabilities in the following way:

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i), \tag{B.13}$$

where t can be freely chosen from the interval $[0, T]$; in particular,

$$P(\boldsymbol{O}|\lambda) = \beta_0(I) = \alpha_T(E). \tag{B.14}$$

The second equality is a direct consequence of the second assumption (see Section B.1.1); moreover, that assumption is also responsible of the following:

$$\alpha_T(i) = 0 \iff s_i \neq E \wedge O_T = \{E\}. \tag{B.15}$$

The number of times state s_i is visited during the generation of the sequence of observable outputs $\boldsymbol{O}$ is defined as:

$$\Gamma_i = \sum_{t=0}^{T-1} \gamma_t(i) \tag{B.16}$$

where (Rabiner, 1989, eq. 27):

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(\boldsymbol{O}|\lambda)}; \tag{B.17}$$

therefore,

$$\Gamma_i = \frac{1}{P(\boldsymbol{O}|\lambda)} \sum_{t=0}^{T-1} \alpha_t(i)\beta_t(i). \tag{B.18}$$

For a complete text, i.e. a text that begins and ends with the end-of-sentence mark, $\alpha_0(i)\beta_0(i) = \alpha_T(i)\beta_T(i) = P(\boldsymbol{O}|\lambda)$; consequently, the sum over time can be rewritten in the following way:

$$\sum_{t=0}^{T-1} \alpha_t(i)\beta_t(i) = \sum_{t=1}^{T} \alpha_t(i)\beta_t(i). \tag{B.19}$$

This result, that will be useful later, can be intuitively understood: the final state is visited twice, once at the beginning of the text and another one at the end, but it must be counted only once. As it does not matter which of them is counted, the sum over the time appearing in Rabiner (1989) can be changed from 1 to T, to 0 to $T-1$.

The following equation defines the expected number of times that the transition from state s_i to state s_j is performed during the generation of the sequence of observable outputs $\boldsymbol{O}$:

$$\Xi_{ij} = \sum_{t=0}^{T-1} \xi_t(i,j), \tag{B.20}$$

where (Rabiner, 1989, eq. 37):

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(\boldsymbol{O}|\lambda)}; \tag{B.21}$$

therefore,

$$\Xi_{ij} = \frac{1}{P(\boldsymbol{O}|\lambda)} \sum_{t=0}^{T-1} \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j). \tag{B.22}$$

The number of times the model emits the word class v_k from hidden state s_j when generating the sequence of observable outputs $\boldsymbol{O}$ is defined as:

$$\Phi_{jk} = \sum_{t=0}^{T-1} \varphi_t(j,k), \tag{B.23}$$

where

$$\varphi_t(j,k) = \frac{\alpha_t(j)\beta_t(j)\delta_{v_k,O_t}}{P(\boldsymbol{O}|\lambda)}; \tag{B.24}$$

that is,

$$\Phi_{jk} = \frac{1}{P(\boldsymbol{O}|\lambda)} \sum_{t=0}^{T-1} \alpha_t(j)\beta_t(j)\delta_{v_k,O_t} \tag{B.25}$$

For debugging purposes it may be useful to notice that:

$$\Gamma_i = \sum_{k=1}^{M} \Phi_{ik} = \sum_{j=1}^{N} \Xi_{ij} \tag{B.26}$$

The computation of Γ_i, Ξ_{ij} and Φ_{jk} requires to process the training text forward to compute each $\alpha_t(i)$ and the likelihood of the whole text $P(\boldsymbol{O}|\lambda)$, and backward to calculate Γ_i, Ξ_{ij} and Φ_{jk} incrementally; thus, only $\beta_{t+1}(j)$ is stored in each iteration.

B.3.4 New parameters

From the previous equations, and after processing the whole corpus, the HMM parameters are updated through the following Baum-Welch general equations:

$$\overline{a_{ij}} = \frac{\Xi_{ij}}{\Gamma_i} \tag{B.27}$$

and

$$\overline{b_j}(k) = \frac{\Phi_{jk}}{\Gamma_j} \tag{B.28}$$

where the initial observable output (O_o) has been added in order to make both denominators equal, in contrast with Equations (40b) and (40c) by Rabiner (1989) in which both denominators are different.

Equations (B.27) and (B.28) update the HMM parameters without performing any smoothing. However, smoothed parameters can be easily obtained through Equations (B.1) and (B.5) as Ξ_{ij} is an estimation of the number of times $n(s_i s_j)$ hidden state s_i is followed by hidden state s_j in the training corpus, Γ_i is an estimation of the number of times $n(s_i)$ hidden state s_i appears in the training corpus, and Φ_{jk} is an estimation of the number of times $n(v_k, s_j)$ the model emits the word class v_k from the hidden state s_j.

B.3.5 Segmentation

When tagging a given text, each time an unambiguous word (with ambiguity class $\{X\}$) appears, the HMM can only be in the state corresponding to the PoS tag of that word (X) as a consequence of the third assumption done in Section B.1.1. This property allows for a more efficient implementation, as suggested by Cutting et al. (1992), because it is not necessary to store the whole text, but the sequence of words

between two unambiguous words (both included) and treat that sequence of words as a whole text.[5] Consider that the training text is segmented in G different segments; each segment g, starting at $t = i_g$ and ending at $t = e_g$, has an initial word belonging to the unambiguous ambiguity class $\{I_g\}$, a final word belonging to the unambiguous ambiguity class $\{E_g\}$ and zero or more ambiguous words between them.

Each of the needed values Ξ_{ij}, Φ_{jk} and Γ_i can be calculated as a sum over all segments:

$$\Xi_{ij} = \sum_{g=1}^{G} \Xi_{ij}^{(g)} \tag{B.29}$$

$$\Phi_{jk} = \sum_{g=1}^{G} \Phi_{jk}^{(g)} \tag{B.30}$$

$$\Gamma_i = \sum_{g=1}^{G} \Gamma_i^{(g)} \tag{B.31}$$

and the computation for each segment can be done as if they were a complete independent text by using only local information to each segment.

Now, the calculation of $\Xi_{ij}^{(g)}$ is explained in detail. The calculation of $\Phi_{jk}^{(g)}$ and $\Gamma_i^{(g)}$ is completely analogous and can be easily inferred. It can be easily proven that:

$$\alpha_{i_g}(i) = P(O_1 \dots O_{i_g})\delta_{i,I_g} \tag{B.32}$$

and that:

$$P(O_1 \dots O_T) = P(O_1 \dots O_{e_g})\beta_{e_g}(E_g) \tag{B.33}$$

Then if we define for $i, j = 1 \dots N$,

$$\alpha_{t-i_g}^{g}(i) = \frac{\alpha_t(i)}{\alpha_{i_g}(I_g)} \tag{B.34}$$

and

$$\beta_{t-i_g}^{g}(j) = \frac{\beta_t(j)}{\beta_{e_g}(E_g)} \tag{B.35}$$

the expression for $\Xi_{ij}^{(g)}$ can be easily derived:

$$\Xi_{ij}^{(g)} = \frac{1}{P^g} \sum_{\tau=0}^{T_g-1} \alpha_\tau^g(i) a_{ij} b_j(O_{i_g+\tau+1}) \beta_{\tau+1}^g(j). \tag{B.36}$$

This equation is analogous to Equation (B.22) but, in this case, P^g is not $P(O_{i_g} \dots O_{e_g})$ because, in general, $I_g \neq E_g$:

$$P^{(g)} = \frac{P(\boldsymbol{O}|\lambda)}{\alpha_{i_g}(I_g)\beta_{e_g}(E_g)} = \alpha_{T_g}^g(E_g) \tag{B.37}$$

[5]This is only valid for a first-order HMM, in n-th order HMMs, segments would be delimited by a sequence of n unambiguous words.

The new forward α^g and backward β^g probabilities for each segment are recursively defined in an analogous way to that of the whole text:

$$\alpha_\tau^g(i) = \left[\sum_{j=1} \alpha_{\tau-1}^g(j) a_{ji}\right] b_i(O_{i_g+\tau}) \tag{B.38}$$

$$\alpha_0^g(i) = \delta_{i,I_g} \tag{B.39}$$

$$\beta_\tau^g(i) = \sum_{j=1} a_{ij} \beta_{\tau+1}^g(j) b_j(O_{i_g+\tau+1}) \tag{B.40}$$

$$\beta_{T_g}^g(i) = 1 \;\; \forall j \in [1, N] \tag{B.41}$$

Analogously Equations (B.18) and (B.25) are transformed into:

$$\Gamma_i^{(g)} = \frac{1}{P^g} \sum_{\tau=0}^{T_g-1} \alpha_\tau^g(i) \beta_\tau^g(i) \tag{B.42}$$

and

$$\Phi_{jk}^{(g)} = \frac{1}{P^g} \sum_{\tau=0}^{T_g-1} \alpha_\tau^g(j) \beta_\tau^g(j) \delta_{v_k, O_{i_g+\tau}}. \tag{B.43}$$

The note at the end of Section B.3.3 about the implementation of the computation of Γ_i, Ξ_{ij} and Φ_{jk} can be easily adapted for the computation of $\Gamma_i^{(g)}$, $\Xi_{ij}^{(g)}$ and $\Phi_{jk}^{(g)}$.

B.3.6 Parameter initialization

The Baum-Welch algorithm is used to iteratively reestimate an HMM whose parameters has been previously estimated or initialized. HMM parameters can be initialized, in absence of knowledge, using the method proposed by Kupiec (1992).

The initialization proposed by Kupiec (1992) consists of estimating from corpora the counts needed to calculate the HMM parameters through Equations (B.1) and (B.5).

The number of times state s_i is followed by state s_j in the training corpus is approximated as follows:

$$\tilde{n}(s_i s_j) = \sum_{v_k : s_i \in v_k} \sum_{v_l : s_j \in v_l} \frac{m(v_k v_l)}{|v_k| \; |v_l|} \tag{B.44}$$

where $m(v_K v_l)$ stands for the number of times observable output v_k is followed by observable output v_l in the training corpus.

The number of times observable output v_k is emitted from state s_j in the training corpus:

$$\tilde{n}(v_k, s_j) = \begin{cases} \frac{m(v_k)}{|v_k|} & \text{if } s_j \in v_k \\ 0 & \text{otherwise} \end{cases} \tag{B.45}$$

where $m(v_k)$ refers to the number of time observable output v_k appears in the training corpus.

Finally, the number of occurrences of hidden state s_i in the training corpus is approximated through the following equation:

$$\tilde{n}(s_i) = \sum_{v_k : s_i \in v_k} \frac{m(v_k)}{|v_k|}. \tag{B.46}$$

B.4 Maximum likelihood estimate method

The *maximum likelihood estimate* (MLE) is a direct method to estimate in supervised manner the HMM parameters. To this end frequency counts $n(\cdot)$ must be collected. Since the training corpus has been disambiguated, each segment has only one disambiguation; it is easy to collect these frequency counts and use them to estimate the smoothed transition and emission probabilities through Equations (B.1) and (B.5).

B.5 Viterbi algorithm

Once the HMM parameters have been estimated, and independently of the method used for training, the Viterbi algorithm (Rabiner, 1989; Manning and Schütze, 1999, p. 332) is used for disambiguation. This dynamic-programming algorithm gives the sequence of PoS tags that maximizes its likelihood given the observable outputs.

Section B.3.5 describes how the input text can be segmented for a more efficient implementation of the Baum-Welch algorithm. The same segmentation can be used to apply the Viterbi algorithm to text segments and treat each segment as a whole text. Remember from that section that each segment consists of a sequence of words starting at $t = i_g$ and ending at $t = e_g$, and that the initial word is an unambiguous word belonging to the unambiguity class $\{I_g\}$ and the final word is also an unambiguous word belonging to the non-ambiguity class $\{E_g\}$.

Now follows the mathematical justification for the Viterbi algorithm using the notation by Rabiner (1989); note that the formulas are slightly modified because they are applied for the disambiguation of text segments. Equation (30) by Rabiner (1989) may be rewritten to define the highest probability along a single path ending in state s_i for the first t observable outputs of text segment $g = O_{i_g} O_{i_g+1} \dots O_t \dots O_{e_g}$:

$$\delta_t(i) = \max_{q_{i_g}, q_{i_g+1}, \dots, q_{t-1}} P(q_{i_g} q_{i_g+1} \dots q_t = s_i, O_{i_g} O_{i_g+1} \dots O_t \mid \lambda), \tag{B.47}$$

and by induction:

$$\delta_{t+1}(j) = \max_i \delta_t(i) a_{ij} b_j(O_{t+1}). \tag{B.48}$$

In order to retrieve the sequence of states for each t and j the sequence of states that maximizes Equation (B.48) is stored in the array $\psi_t(j)$. The complete dynamic-programming algorithm to retrieve the best sequence of states consists of the following steps:

1. Initialization:
$$\delta_{i_g}(i) = \delta_{I_g,s_i}, \quad 1 \leq i \leq N \tag{B.49}$$
where δ_{I_g,s_i} is the Kronecker delta defined in Section B.3.1 (Equation B.8).

2. Induction:
$$\delta_t(j) = \max_{1\leq i\leq N}(\delta_{t-1}(i)a_{ij})b_j(O_t), \quad \begin{array}{l} i_g + 1 \leq t \leq e_g \\ 1 \leq j \leq N \end{array} \tag{B.50}$$
$$\psi_t(j) = \arg\max_{1\leq i\leq N}(\delta_{t-1}(i)a_{ij}), \quad \begin{array}{l} i_g + 1 \leq t \leq e_g \\ 1 \leq j \leq N \end{array} \tag{B.51}$$

3. Termination:
$$P^* = \max_{1\leq i\leq N}(\delta_{e_g}(i)) \tag{B.52}$$
$$q^*_{e_g} = \arg\max_{1\leq i\leq N}(\delta_{e_g}(i)) = E_g \tag{B.53}$$

4. Retrieval of the state sequence:
$$q^*_t = \psi_{t+1}(q^*_{t+1}), \quad t = e_g - 1, e_g - 2, \ldots, i_g + 1. \tag{B.54}$$
Note that when retrieving the best state sequence, the state q_{i_g} corresponding to the first word of segment g is ignored. This is because each segment is started with an unambiguous word that is the same word ending the previous segment. This is true for all segments, including the first one, due to the first assumption (Section B.1.1).

Appendix C

Open-source software released as part of this thesis

All the methods and techniques described in this thesis have been released under open-source licenses in order to ensure the reproducibility of all the experiments conducted, and to allow other researchers to use and improve them. This appendix briefly overviews the open-source software released and relates each software package with the experiments conducted in each chapter.

C.1 `apertium-tagger-training-tools`

The MT-oriented method described in Chapter 2 to train an SL HMM-based PoS tagger by using information from the TL and from the rest of modules of the MT engine in which the resulting SL PoS tagger is to be embedded, is implemented inside package `apertium-tagger-training-tools`, which is released under the GNU GPL license version 2; it can be freely downloaded from `http://sf.net/projects/apertium`. This package also implements the pruning method described in Chapter 3, and the clustering of PoS tags described in Chapter 4.

After executing the program `apertium-tagger-tl-trainer` with the needed parameters a file containing the HMM parameters is produced; this file can be directly used within the Apertium MT platform (see appendix A). This simplifies the building of an Apertium-based MT system for a new pair of languages. The path pruning strategy described in Chapter 3 may also be activated by using the appropriate parameters.

The program that runs the clustering algorithm (`apertium-tagger-tagset-clustering`) receives as input a file containing the HMM parameters previously obtained by means of any HMM training method, and a distance threshold. After clustering a

text file defining how the fine-grained tags must be grouped into coarser tags is produced. This file can be used through the `apertium-tagger-states-merging` program to disambiguate new corpora; note, however, that `apertium-tagger-states-merging` only provides a proof-of-concept implementation.

C.2 `apertium-transfer-tools`

The method described in Chapter 5 to automatically infer shallow-transfer MT rules from parallel corpora is implemented in package `apertium-transfer-tools`, which has been also released under the GNU GPL license version 2; it can be downloaded from `http://sf.net/projects/apertium`. This package provides a set of tools which allows for the generation of transfer rules in the Apertium (XML) format; these rules can be directly used by the Apertium MT platform.

Although this package is aimed at the generation of Apertium transfer rules it can easily be adapted to generate shallow-transfer rules for other MT platforms; moreover, some of the tools it provides can be used for other purposes such as the extraction of bilingual phrase pairs or the symmetrization of previously computed alignments. This package depends on the open-source GIZA++ toolkit[1] (Och and Ney, 2003) to compute word alignments; nevertheless, it can be easily adapted to use other aligners such as LIHLA (Caseli et al., 2005).

[1] `http://code.google.com/p/giza-pp`

Index of abbreviations

Index of frequently used symbols

List of Figures

List of Tables

Bibliography

Agirre, E. and Edmonds, P., editors (2007). *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*. Springer.

Armentano-Oller, C., Carrasco, R. C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., Sánchez-Martínez, F., and Scalco, M. A. (2006). Open-source Portuguese-Spanish machine translation. In *Computational Processing of the Portuguese Language, Proceedings of the 7th International Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR 2006*, volume 3960 of *Lecture Notes in Computer Science*, pages 50–59. Springer-Verlag, Itatiaia, Brazil.

Armentano-Oller, C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Bonev, B., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., and Sánchez-Martínez, F. (2005). An open-source shallow-transfer machine translation toolbox: consequences of its release and availability. In *OSMaTran: Open-Source Machine Translation, A workshop at Machine Translation Summit X*, pages 23–30, Phuket, Thailand.

Armentano-Oller, C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Montava Belda, M. A., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., and Sánchez-Martínez, F. (2007). Apertium, una plataforma de código abierto para el desarrollo de sistemas de traducción automática. In *Proceedings of the FLOSS International Conference 2007*, pages 5–20, Jerez de la Frontera, Spain.

Armentano-Oller, C. and Forcada, M. L. (2006). Open-source machine translation between small languages: Catalan and aranese occitan. In *Proceedings of Strategies for developing machine translation for minority languages (5th SALTMIL workshop on Minority Languages)*, pages 51–54, Genoa, Italy.

Armstrong, S., Flanagan, M., Graham, Y., Groves, D., Mellebeek, B., Morrissey, S., Stroppa, N., and Way, A. (2006). Matrex: Machine translation using examples. In *TC-STAR OpenLab Workshop on Speech Translation*, Trento, Italy.

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563.

Bernardo, J. M. and Rueda, R. (2002). Bayesian hypothesis testing: A reference approach. *International Statistical Review*, 70:351–372.

Brants, T. (1995a). Estimating HMM topologies. In *Tbilisi Symposium on Language, Logic, and Computation*, Tbilisi, Republic of Georgia.

Brants, T. (1995b). Tagset reduction without information loss. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 287–289, Cambridge, MA, USA.

Brants, T. (1996). Estimating Markov model structures. In *Proceeding of the 4th International Conference on Spoken Language Processing (ICSLP'96)*, volume 2, pages 893–896, Philadelphia, PA. USA.

Brants, T. and Samuelsson, C. (1995). Tagging the Teleman corpus. In *Proceedings of the 10th Nordic Conference of Computational Linguistics*, Helsinki, Finland.

Brill, E. (1992). A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy.

Brill, E. (1995a). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.

Brill, E. (1995b). Unsupervised learning of disambiguation rules for part of speech tagging. In Yarowsky, D. and Church, K., editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 1–13, Cambridge, MA, USA.

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

Brown, R. D. (1999). Adding linguistic knowledge to a lexical example-based translation system. In *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pages 22–32, Chester, UK.

Canals, R., Esteve, A., Garrido, A., Guardiola, M. I., Iturraspe-Bellver, A., Montserrat, S., Pérez-Antón, P., Ortiz, S., Pastor, H., and Forcada, M. L. (2000). internostrum: a spanish-catalan machine translation system. *Machine Translation Review*, (11):21–25.

Canals-Marote, R., Esteve-Guillen, A., Garrido-Alenda, A., Guardiola-Savall, M., Iturraspe-Bellver, A., Montserrat-Buendia, S., Ortiz-Rojas, S., Pastor-Pina, H., Perez-Antón, P. M., and Forcada, M. L. (2001). The Spanish-Catalan machine translation system interNOSTRUM. In *Proceedings of MT Summit VIII: Machine Translation in the Information Age*, pages 73–76, Santiago de Compostela, Spain.

Carbonell, J., Klein, S., Miller, D., Steinbaum, M., Grassiany, T., and Frei, J. (2006). Context-based machine translation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, "Visions for the Future of Machine Translation"*, pages 19–28, Cambridge, MA, USA.

Carl, M. and Way, A., editors (2003). *Recent Advances in Example-Based Machine Translation*, volume 21. Springer.

Caseli, H. M., Nunes, M. G. V., and Forcada, M. L. (2005). LIHLA: A lexical aligner based on language-independent heuristics. In *Anais do V Encontro Nacional de Inteligência Artificial (ENIA 2005)*, pages 641–650, São Leopoldo-RS, Brazil.

Caseli, H. M., Nunes, M. G. V., and Forcada, M. L. (2006). Automatic induction of bilingual resources from aligned parallel corpora: application to shallow-transfer machine translation. *Machine Translation*, 20(4):227–245. Published in 2008.

Cicekli, I. and Güvenir, H. A. (2001). Learning translation templates from bilingual translation examples. *Applied Intelligence*, 15(1):57–76.

Corbí-Bellot, A. M., Forcada, M. L., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Alegria, I., Mayor, A., and Sarasola, K. (2005). An open-source shallow-transfer machine translation engine for the Romance languages of Spain. In *Proceedings of the 10th European Associtation for Machine Translation Conference*, pages 79–86, Budapest, Hungary.

Cutting, D., Kupiec, J., Pedersen, J., and Sibun, P. (1992). A practical part-of-speech tagger. In *Third Conference on Applied Natural Language Processing. Association for Computational Linguistics. Proceedings of the Conference.*, pages 133–140, Trento, Italy.

Dermatas, E. and Kokkinakis, G. (1995). Automatic stochastic tagging of natural language texts. *Computational Linguistics*, 21(2):137–163.

Dien, D. and Kiem, H. (2003). Pos-tagger for English-Vietnamese bilingual corpus. In *Proceedings of HLT-NAACL 2003 Workshop: Building and Using Parallel Texts, Data Driven Machine Translation and Beyond*, pages 88–95, Edmonton, AB, Canada.

Dirix, P., Schuurman, I., and Vandeghinste, V. (2005). Metis II: Example-based machine translation using monolingual corpora - system description. In *Proceedings of the 2nd Workshop on Example-Based Machine Translation*, pages 43–50, Phuket, Tailand.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience. Second edition.

Dugast, L., Senellart, J., and Koehn, P. (2007). Statistical post-editing on SYSTRAN's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223, Prague, Czech Republic.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the Bootstrap*. CRC Press.

Forcada, M. L. (2006). Open-source machine translation: an opportunity for minor languages. In *Proceedings of Strategies for developing machine translation for minority languages (5th SALTMIL workshop on Minority Languages)*, pages 51–54, Genoa, Italy.

Forcada, M. L., Bonev, B. I., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Armentano-Oller, C., Montava, M. A., and Tyers, F. M. (2007). Documentation of the open-source shallow-transfer machine translation platform Apertium. `http://xixona.dlsi.ua.es/~fran/apertium2-documentation.pdf`.

Gale, W. A. and Church, K. W. (1990). Poor estimates of context are worse than none. In *Proceedings of a workshop on Speech and Natural Language*, pages 283–287, Hidden Valley, PA, USA.

Gale, W. A. and Sampson, G. (1995). Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3):217–237.

Garrido-Alenda, A., Forcada, M. L., and Carrasco, R. C. (2002). Incremental construction and maintenance of morphological analysers based on augmented letter transducers. In *Proceedings of 9th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 53–62, Keihanna, Japan.

Garrido-Alenda, A., Zarco, P. G., Pérez-Ortiz, J. A., Pertusa-Ibáñez, A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Scalco, M. A., and Forcada, M. L. (2004). Shallow parsing for Portuguese-Spanish machine translation. In Branco, A., Mendes, A., and Ribeiro, R., editors, *Language Technology for Portuguese: shallow processing tools and resources*, pages 135–144. Edições Colibri, Lisbon, Portugal.

Gilabert-Zarco, P., Herrero-Vicente, J., Ortiz-Rojas, S., Pertusa-Ibáñez, A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Samper-Asensio, M., Scalco, M. A., and Forcada, M. L. (2003). Construcciǿn rápida de un sistema de traducciǿn automática español-portugués partiendo de un sistema español-catalán. In *XIX Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural*, pages 279–284, Alcalá de Henares, Spain.

Gough, N. and Way, A. (2004). Robust large-scale EBMT with marker-based segmentation. In *Proceedings of the Tenth Conference on Theoretical and Methodological Issues in Machine Translation*, pages 95–104, Baltimore, MD, USA.

Green, T. (1979). The necessity of syntax markers. Two experiments with artificial languages. *Journal of Verbal Learning and Behavior*, 18:481–496.

Grosse, I., Bernaola-Galván, P., Carpena, P., Román-Roldán, R., Oliver, J., and Standley, H. E. (2002). Analysis of symbolic sequences using the Jensen-Shannon divergence. *Physical Review E*, 65(4):041905.

Groves, D. and Way, A. (2005). Hybrid example-based SMT: the best of both worlds? In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 183–190, Ann Arbor, MI, USA.

Hutchins, W. J. and Somers, H. L. (1992). *An Introduction to Machine Translation*. Academic Press, London, UK.

Ide, N. and Véronis, J. (1998). Word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–41.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. The MIT Press.

Kaji, H., Kida, Y., and Morimoto, Y. (1992). Learning translation templates from bilingual text. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 672–678, Nantes, France.

Kim, J. D., Lee, S. Z., and Rim, H. C. (1999). HMM specialization with selective lexicalization. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 121–127, College Park, MD, USA.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Knight, K. (1999). A statistical machine translation tutorial workbook. `http://www.isi.edu/natural-language/mt/wkbk.rtf`. 35 pages.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain.

Koehn, P. and Hoang, H. (2007). Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876, Prague, Czech Republic.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

Kupiec, J. (1992). Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6(3):225–242.

Lavie, A. (2008). Stat-XFER: A general search-based syntax-driven framework for machine translation. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2008)*, pages 362–375, Haifa, Israel.

Lavie, A., Probst, K., Peterson, E., Vogel, S., Levin, L., Font-Llitjós, A., and Carbonell, J. (2004). A trainable transfer-based machine translation approach for languages with limited resources. In *Proceedings of Workshop of the European Association for Machine Translation (EAMT-2004)*, pages 26–27, Valletta, Malta.

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848. English translation in Soviet Physics Doklady, 10(8), 707–710, 1966.

Liu, Y. and Zong, C. (2004). The technical analysis on translation templates. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics (SMC)*, pages 4799–4803, The Hague, The Netherlands.

Lopez, A. (2008). Statistical machine translation. *ACM Computing Surveys*, 40(3):1–49.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–171.

Mikheev, A. (1996). Unsupervised learning of word-category guessing rules. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 327–333, New York, USA.

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

Och, F. J. (1999). An efficient method for determining bilingual word classes. In *EACL'99: Ninth Conference of the European Chapter of the Association for Computational Lingustics*, pages 71–76, Bergen, Norway.

Och, F. J. (2002). *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. PhD thesis, RWTH Aachen University, Aachen, Germany.

Och, F. J. (2005). Statistical machine translation: Foundations and recent advances. Tutorial at MT Summit X (`http://www.mt-archive.info/MTS-2005-Och.pdf`), Phuket, Thailand.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Lingustics*, pages 295–302, Philadelphia, PA, USA.

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

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

Oepen, S., Velldal, E., Lønning, J. T., Meurer, P., and Rosen, V. (2007). Towards hybrid quality-oriented machine translation. On linguistics and probabilities in MT. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation*, pages 144–153, Skövde, Sweden.

Omohundro, S. M. (1992). Best-first model merging for dynamic learning and recognition. In Moody, J. E., Hanson, S. J., and Lippmann, R. P., editors, *Advances in Neural Information Processing Systems*, volume 4, pages 958–965, Denver, CO, USA. Morgan Kaufmann Publishers, Inc.

Ortiz-Rojas, S., Forcada, M. L., and Ramírez-Sánchez, G. (2005). Construcción y minimización eficiente de transductores de letras a partir de diccionarios con paradigmas. In *Procesamiento del Lenguaje Natural nº 35, (XXI Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural)*, pages 51–57, Granada, Spain.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (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, USA.

Pérez-Ortiz, J. A. (2002). *Modelos predictivos basados en redes neuronales recurrentes de tiempo discreto*. PhD thesis, Departament de Llenguatges i Sistemes informàtics, Universitat d'Alacant, Alacant, Spain.

Pla, F. and Molina, A. (2004). Improving part-of-speech tagging using lexicalized HMMs. *Journal of Natural Language Engineering*, 10(2):167–189.

Probst, K., Levin, L., Peterson, E., Lavie, A., and Carbonell, J. (2002). MT for minority languages using elicitation-based learning of syntactic transfer rules. *Machine Translation*, 17(4):245–270.

Purnhagen, H. (1994). *N*-best search methods applied to speech recognition. Master's thesis, Universitetet i Trondheim, Trondheim, Norway.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Ramírez-Sánchez, G., Sánchez-Martínez, F., Ortiz-Rojas, S., Pérez-Ortiz, J. A., and Forcada, M. L. (2006). Opentrad apertium open-source machine translation system: an opportunity for business and research. In *Proceeding of Translating and the Computer 28 Conference*, London, UK.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In Brill, E. and Church, K., editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, Philadephia, PA, USA.

Ratnaparkhi, A. (1998). *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.

Rivlin, Z., Sankar, A., and Bratt, H. (1997). HMM state clustering across allophone class boundaries. In *Proceedings of Eurospeech '97*, pages 127–130, Rhodes, Greece.

Roche, E. and Schabes, Y. (1997). *Finite-State language Processing*, chapter Introduction, pages 1–65. MIT Press.

Sánchez-Martínez, F., Armentano-Oller, C., Pérez-Ortiz, J. A., and Forcada, M. L. (2007a). Training part-of-speech taggers to build machine translation systems for less-resourced language pairs. In *Procesamiento del Lenguaje Natural (XXIII Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural)*, volume 39, pages 257–264, Sevilla, Spain.

Sánchez-Martínez, F. and Forcada, M. L. (2007). Automatic induction of shallow-transfer rules for open-source machine translation. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007)*, volume 2007:1, pages 181–190, Skövde, Sweden. Skövde University Studies in Informatics.

Sánchez-Martínez, F. and Forcada, M. L. (2009). Inferring shallow-transfer machine translation rules from small parallel corpora. *Journal of Artificial Intelligence Research*, 34:605–635.

Sánchez-Martínez, F. and Ney, H. (2006). Using alignment templates to infer shallow-transfer machine translation rules. In *Advances in Natural Language Processing, Proceedings of 5th International Conference on Natural Language Processing FinTAL*, volume 4139 of *Lecture Notes in Computer Science*, pages 756–767. Springer-Verlag, Turku, Finland.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2004a). Cooperative unsupervised training of the part-of-speech taggers in a bidirectional machine translation system. In *Proceedings of the Tenth Conference on Theoretical and Methodological Issues in Machine Translation*, pages 135–144, Baltimore, MD USA.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2004b). Exploring the use of target-language information to train the part-of-speech tagger of machine translation systems. In *Advances in Natural Language Processing, Proceedings of 4th International Conference EsTAL*, volume 3230 of *Lecture Notes in Computer Science*, pages 137–148. Springer-Verlag, Alicante, Spain.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2005). Target-language-driven agglomerative part-of-speech tag clustering for machine translation. In *Proceedings of the International Conference RANLP - 2005 (Recent Advances in Natural Language Processing)*, pages 471–477, Borovets, Bulgaria.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2006). Speeding up target-language driven part-of-speech tagger training for machine translation. In *MICAI 2006: Advances in Artificial Intelligence, Proceedings of the 5th Mexican International Conference on Artificial Intelligence*, volume 4293 of *Lecture Notes in Computer Science*, pages 844–854. Springer-Verlag, Apizaco, Mexico.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2007b). Integrating corpus-based and rule-based approaches in an open-source machine translation system. In *Proceedings of METIS-II Workshop: New Approaches to Machine Translation, a workshop at CLIN 17 - Computational Linguistics in the Netherlands*, pages 73–82, Leuven, Belgium.

Sánchez-Martínez, F., Pérez-Ortiz, J. A., and Forcada, M. L. (2008). Using target-language information to train part-of-speech taggers for machine translation. *Machine Translation*, 22(1-2):29–66.

Sánchez-Villamil, E., Forcada, M. L., and Carrasco, R. C. (2004). Unsupervised training of a finite-state sliding-window part-of-speech tagger. In *Advances in Natural Language Processing, Proceedings of 4th International Conference EsTAL*, volume 3230 of *Lecture Notes in Computer Science*, pages 454–463. Springer-Verlag, Alicante, Spain.

Schwartz, R. and Chow, Y.-L. (1990). The N-best algorithm: an efficient and exact procedure for finding the n most likely sentece hypotheses. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 81–84.

Simard, M., Ueffing, N., Isabelle, P., and Kuhn, R. (2007). Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206, Prague, Czech Republic.

Somers, H., editor (2003). *Computers and Translation: A translator's guide*, chapter Why translation is difficult for computers (by D. Arnold). Benjamins Translation Library.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (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)*, Genoa, Italy.

Stolcke, A. and Omohundro, S. M. (1994). Best-first model merging for hidden Markov model induction. Technical Report TR-94-003, University of California, Berkeley, CA, USA.

Tinsley, J., Ma, Y., Ozdowska, S., and Way, A. (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, USA.

Vogel, S., Ney, H., and Tillmann, C. (1996). HMM-based word alignment in statistical translation. In *COLING '96: The 16th International Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark.

Webb, A. (2002). *Statistical Pattern Recognition*. Wiley. Second edition.

Yarowsky, D. and Ngai, G. (2001). Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of The Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, pages 200–207, Pittsburgh, PA, USA.

Zens, R., Och, F. J., and Ney, H. (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. Springer-Verlag, Aachen, Germany.