UNIVERSIDAD POLITÉCNICA DE VALENCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Novel statistical approaches to text classification, machine translation and computer-assisted translation

Thesis
presented by Jorge Civera Saiz
supervised by Dr. Alfons Juan Císcar and Dr. Francisco Casacuberta Nolla

May 22, 2008

Novel statistical approaches to text classification, machine translation and computer-assisted translation

Jorge Civera Saiz

Thesis performed under the supervision of doctors
Alfons Juan Císcar and Francisco Casacuberta Nolla
and presented at the Universidad Politécnica de Valencia
in partial fulfilment of the
of the requirements for the degree
Doctor en Informática

Valencia, May 22, 2008

Work partially supported by the EC (FEDER) and the Spanish *Ministerio de Educación y Ciencia* under the MIPRCV ”Consolider Ingenio 2010” research programme (CSD2007-00018), the iTransDoc research project (TIN2006-15694-CO2-01), and the FPU fellowship AP2003-0342

ACKNOWLEDGEMENTS

Posiblemente sean estas líneas de las más leídas en esta tesis, aunque esperemos inocentemente que no se así. Son estas líneas las más agradecidas de escribir, pero a la vez las más difíciles para no olvidar a aquellos que han contribuido de una forma u otra a que esta tesis sea una realidad.

En primer lugar, quisiera agradecer el apoyo siempre incondicional de mi familia en mi vida, y la elaboración de esta tesis no iba a ser una excepción. Ella ha estado siempre ahí cuando lo he necesitado, compartiendo conmigo el sacrificio personal que requiere una tesis, y preocupándose por mí cuando yo no lo he hecho. Sinceramente, esta tesis es tanto mía como suya. De igual forma, quisiera aprovechar estas palabras de agradecimiento, para hacer una especial mención de aquellos familiares que tanto les hubiera gustado compartir este momento con su nieto, pero que por ley de vida no ha podido ser.

En el aspecto académico, como olvidarme de aquellos que han contribuido de forma tan decisiva al contenido de esta tesis, y por supuesto a mi formación investigadora. Mi más profundo agradecimiento a Alfons por compartir conmigo muchas de las ideas que han nutrido y modelado esta tesis, por dedicar tantas horas de su tiempo a mi formación trabajando codo con codo conmigo, y por la confianza depositada y el apoyo económico prestado en esta última fase de mi tesis. También estoy en deuda con Paco por acogerme como becario FPU durante casi cuatro años, y darme la libertad de explorar las diferentes líneas de investigación que han desembocado en esta tesis. Por último, agradecer a Enrique junto con Paco la oportunidad que me brindaron de entrar en este grupo de investigación, trabajar en un proyecto europeo como TT2, y por contribuir a mi formación a través de discusiones, revisiones y correcciones de mi trabajo.

Tampoco puedo olvidar a mis compañeros de fatigas del proyecto TT2, Elsa, Antonio y Luis. Con ellos he pasado muy buenos momentos juntos, compartiendo viajes de los que guardo anécdotas y recuerdos imborrables que darían para escribir un libro. Ellos han contribuido a mi tesis con su dedicación en el proyecto TT2, durante los tres años que duró, y parte del trabajo aquí presentado es también el suyo.

Asimismo, agradecer el apoyo prestado por mis otros compañeros del grupo PRHLT en el ITI y en el DSIC. Especialmente, a Jesús Andrés por sus explicaciones, sugerencias y correcciones que ya son parte de esta tesis, a Daniel Ortiz, Vicent Alabau y Germán por las interesantes discusiones mantenidas, a Alejandro por cederme la plantilla de esta tesis que tanto trabajo le dió en su momento, y al maestro José Ramón por estar siempre a mi lado en el nuevo ITI. Igualmente, aprovechar para pedir disculpas a aquellos que en estos últimos meses de la elaboración de mi tesis hubieran necesitado de mi ayuda.

No puedo tampoco olvidarme de mis compañeros de la facultad con los que he compartido tantas horas de clase durante los cuatros años que estuve con ellos. Algunos de ellos, como Germán Moltó, Miguel Masmano y Sergio Grau se han convertido en amigos que aunque la vida nos ha llevado por caminos diferentes, como bien dice Germán, “el recuerdo permanece”.

A todos los que han contribuido a esta tesis y me han apoyando, mi más sincero agradecimiento.

Jorge Civera Saiz
Valencia, May 22, 2008

ABSTRACT

This thesis presents diverse contributions in the fields of text classification, machine translation and computer-assisted translation under the statistical framework.

In text classification, a new application called bilingual text classification is proposed together with a series of models to capture bilingual information. To this purpose two main approaches were presented, the first of them is based on a naive crosslingual-independent assumption and the second, on a more sophisticated crosslingual word-correlation framework. As far as the naive assumption is concerned, five unigram models and smoothed n-gram languages models are introduced. These models were evaluated on three tasks of increasing complexity, considering the most complex of these tasks under the viewpoint of a bilingual machine-aided indexing application. The crosslingual word-correlation framework is represented by bilingual models that integrate a translation model. In our case this model is the well-known M1[a] translation model in conjunction with a unigram model. This model was tested on two of the simpler previously mentioned tasks superseding the naive approximation.

In machine translation, the statistical word-alignment translation models M1, M2[b] and HMM are extended under the mixture modelling approach in order to define context-specific translation models. In the case of the M2 model, a mixture extension of an already existing iterative dynamic-programming search algorithm for the M2 model is also defined. This search algorithm allows us to directly assess the translation quality of the M2 mixture model on a semi-artificial controlled task, obtaining statistically significant improvements over the conventional M2 model. Moreover, an extensive experimental evaluation of these three models is carried out on two well-known shared tasks. These two tasks are used to assess on the one hand, the quality of the alignments obtained as a byproduct of the M1, M2 and HMM models and on the other hand, the translation quality of a statistical phrase-based system seeded with these alignments. As a result of this evaluation we observed that the M2 mixture model offered statistically significant betterment in alignment quality with respect to the conventional M2 model. In addition, the evaluation of translation quality brought to light slight, but systematic improvements in translation quality for all three models, achieving state-of-the-art results for the HMM mixture model.

Finally, an interactive and predictive computer-assisted translation system based on stochastic finite-state transducers is presented. This system integrates well-known efficient error-correcting and n-best parsing algorithms that are adapted

[a] Known as IBM 1 model in the literature.

[b] Known as IBM 2 model in the literature.

and implemented in order to guarantee low response time, while preserving adequate translation quality. The system was automatically tested on two corpora devoted to technical user manuals and bulletins of the European Union. The former corpus served as a bedtest for a thoroughly manual evaluation performed by translation agencies involved in the European project TransType2. Both, automatic and manual evaluations reported a significant reduction in typing effort, speeding up the translation process, and achieving so, the final goal of computer-assisted translation systems.

Resumen

Esta tesis presenta diversas contribuciones en los campos de la clasificación automática de texto, traducción automática y traducción asistida por ordenador bajo el marco estadístico.

En clasificación automática de texto, se propone una nueva aplicación llamada clasificación de texto bilingüe junto con una serie de modelos orientados a capturar dicha información bilingüe. Con tal fin se presentan dos aproximaciones a esta aplicación; la primera de ellas se basa en una asunción naive que contempla la independencia entre las dos lenguas involucradas, mientras que la segunda, más sofisticada, considera la existencia de una correlación entre palabras en diferentes lenguas. La primera aproximación dió lugar al desarrollo de cinco modelos basados en modelos de unigrama y modelos de n-gramas suavizados. Estos modelos fueron evaluados en tres tareas de complejidad creciente, siendo la más compleja de estas tareas analizada desde el punto de vista de un sistema de ayuda a la indexación de documentos. La segunda aproximación se caracteriza por modelos de traducción capaces de capturar correlación entre palabras en diferentes lenguas. En nuestro caso, el modelo de traducción elegido fue el modelo M1[c] junto con un modelo de unigramas. Este modelo fue evaluado en dos de las tareas más simples superando la aproximación naive, que asume la independencia entre palabras en differentes lenguas procedentes de textos bilingües.

En traducción automática, los modelos estadísticos de traducción basados en palabras M1, M2[d] y HMM son extendidos bajo el marco de la modelización mediante mixturas, con el objetivo de definir modelos de traducción dependientes del contexto. Asimismo se extiende un algoritmo iterativo de búsqueda basado en programación dinámica, originalmente diseñado para el modelo M2, para el caso de mixturas de modelos M2. Este algoritmo de búsqueda nos permite evaluar directamente la calidad de la traducción del modelo de mixturas de M2 en una tarea controlada y semiartificial, obteniendo mejoras estadísticamente significativas sobre el modelo M2 convencional. Además, estos tres modelos fueron sometidos a una amplia evaluación experimental llevada a cabo en dos tareas de referencia para la comunidad de traducción automática estadística. Estas dos tareas fueron utilizadas para evaluar la calidad de los alineamientos de estos modelos, así como la calidad de sus traducciones de forma indirecta. Los alineamientos fueron obtenidos como subproducto de los modelos M1, M2 y HMM, mientras que las traducciones fueron generadas por un sistema de traducción estadística basado en segmentos bilingües obtenidos a partir de estos alineamientos. Como resultado de esta eva-

[c] Conocido como modelo 1 de IBM en la literatura.

[d] Conocido como modelo 2 de IBM en la literatura.

luación, se obtuvieron mejoras que son estadísticamente significativas en la calidad de los alineamientos del modelo de mixturas de modelos M2 respecto al modelo M2 convencional. La evaluación de la calidad de la traducción desveló mejoras menores, pero sistemáticas en la calidad de la traducciones ofrecidas por estos tres modelos, logrando resultados a la altura del estado del arte para el modelo de mixturas de HMM.

Por último, presentamos un sistema interactivo y predictivo de ayuda a la traducción basado en transductores estocásticos de estados finitos. Este sistema integra algoritmos de análisis eficientes para la corrección de errores y el cálculo de las mejores traducciones, que son adaptados e implementados para garantizar un tiempo de respuesta bajo, a la vez que se preserva una calidad de traducción adecuada. El sistema presentado fue evaluado automáticamente en dos corpora, uno de ellos consistente en una colección de manuales técnicos de usuario, y el otro formado por boletines de la Unión Europea. El primero de los corpora fue utilizado para una evaluación manual por agencias de traducción en el marco del proyecto europeo TransType2. Tanto la evaluación manual como la automática proporcionaron reducciones significativas en el esfuerzo necesario para traducir dichos textos, acelerando el proceso de traducción, y consiguiendo de esta forma el objetivo final de los sistema de ayuda a la traducción.

RESUM

Aquesta tesi presenta diverses contribucions als camps de la classificació automàtica de text, traducció automàtica i traducció assistida per ordinador sota el marc estadístic.

En classificació automàtica de text, es proposa una nova aplicació anomenada classificació de text bilingüe juntament amb una sèrie de models orientats a capturar aquesta informació bilingüe. Amb aquest fi es presenten dues aproximacions a aquesta aplicació; la primera d'elles es fonamenta en una assumpció naive que contempla la independència entre les dues llengües involucrades, mentre que la segona, més sofisticada, considera l'existència d'una correlació entre paraules en diferents llengües. La primera aproximació donà lloc al desenvolupament de cinc models fonamentats en models d'unigrama i models de llenguatge de n-grames suavitzats. Aquests models van ser avaluats en tres tasques de complexitat creixent, sent la més complexa d'aquestes tasques analitzada des del punt de vista d'un sistema d'ajuda a la indexació de documents. La segona aproximació es caracteritza per models de traducció capaços de capturar la correlació entre paraules en diferents llengües. En el nostre cas, el model de traducció elegit va ser el model M1[e] juntament amb un model d'unigrames. Aquest model va ser avaluat en dos de les tasques més simples superant l'aproximació naive, que assumeix la independència entre paraules en differents llengües procedents de textos bilingües.

En traducció automàtica, els models estadístics de traducció basats en paraules M1, M2[f] i HMM són estesos sota el marc de la modelització mitjançant mixtures, amb l'objectiu de definir models de traducció dependents del context. En el cas del model M2, també es va estendre per al cas de mixtures un algorisme de cerca iteratiu basat en programació dinàmica per a aquest model. Aquest algorisme de cerca ens permet avaluar directament la qualitat de la traducció del model de mixtures de M2 en una tasca controlada i semiartificial, obtenint millores estadísticament significatives sobre el model M2 convencional. A més a més, aquests tres models van ser sotmesos a una àmplia avaluació experimental portada a terme en dues tasques de referència per a la comunitat de traducció automàtica estadística. Aquestes dues tasques van ser utilitzades per avaluar d'una banda, la qualitat dels alineaments obtinguts com a subproducte dels models M1, M2 i HMM i d'altra banda, la qualitat de les traduccions d'un sistema de traducció estadística basat en segments generat a partir d'aquests alineaments. Com a resultat d'aquesta avaluació, es van obtenir millores significatives en la qualitat dels alineaments del model de mixtures de M2 respecte al model M2 convencional. L'avaluació de la qualitat

[e] Conegut com model 1 d'IBM en la literatura.

[f] Conegut com model 2 d'IBM en la literatura.

de la traducció va desvetlar millores menors, però sistemàtiques en la qualitat de la traduccions oferides per aquests tres models, aconseguint resultats a l'altura de l'estat de l'art per al model de mixtures de HMM.

Finalment, vam presentar un sistema interactiu i predictiu d'ajuda a la traducció basat en transductors estocàstics d'estats finits. Aquest sistema integra algorismes d'anàlisi eficients per a la correcció d'errors i el càlcul de les millors traduccions, que són adaptats i implementats per garantir un temps de resposta baix, alhora que es preserva una qualitat de traducció adequada. El sistema presentat va ser avaluat automàticament en dos corpora, un d'ells consistent en una collecció de manuals tècnics d'usuari, i l'altre format per butlletins de la Unió Europea. El primer dels corpora va ser utilitzat per a una avaluació manual portada a terme per agències de traducció al marc del projecte europeu TransType2. Tant l'avaluació manual com l'automàtica, van proporcionar reduccions significatives en l'esforç necessari per traduir aquests textos, accelerant el procés de traducció, i aconseguint, d'aquesta forma, l'objectiu final dels sistemes d'ajuda a la traducció.

PREFACE

Natural language processing (NLP) is an hectic research field that aims at developing computer systems able to automatically generate and understand natural human language, both written and spoken. NLP is a subfield of artificial intelligence and linguistics, and as such it tends to combine theories, methodologies and experts coming from both worlds in order to address challenging problems that sometimes require world knowledge to be successfully solved. This thesis explores two important areas of NLP: text classification (TC) and machine translation (MT).

The purpose of TC is to convert an unstructured repository of documents into a structured one by automatically assigning documents to a predefined number of groups, in the case of text clustering, or to a set of predefined categories, in the case of text categorisation. Doing so, the task of storing, searching and browsing documents in these repositories is significantly simplified. These days TC technology seems to have reached a mature stage, however there are still open problems and challenges ahead.

Among these open problems and challenges we find the classification of multilingual documents. Multilingual documentation is a common phenomenon in many official institutions (EU parliament, the Canadian Parliament, UN sessions, Catalan and Basque Parliaments in Spain, etc.) and private companies (user's manuals, newspapers, books, etc.). In many cases, this textual information needs to be categorised by hand, entailing a time-consuming and arduous burden. In this thesis we focused on the classification of bilingual documents, since it is closely related to MT in which bilingual parallel texts are widely used to train translation systems.

Bilingual TC is a novel application in the field of TC strongly characterised by word correlation across languages. This word correlation comes from the fact that bilingual texts are mutual translations. Given the latter scenario, we propose two main approaches to tackle bilingual TC. First, we may naively consider that bilingual texts were generated independently and therefore, there is not exist any crosslingual relation between words found in mutual translations. Alternatively. we can realistically assume that an underlying crosslingual word mapping exists and can be exploited to boost our bilingual classifier. Undoubtedly, the latter approach is significantly more complex than the former, however the crosslingual structure apprehended by the latter is a valuable information that cannot be neglected.

The other area in NLP to which this thesis is devoted is MT. MT is the use of computers to automate the translation of texts or utterances from one language into another language, while the underlying meaning remains the same. Current MT technology is focused on three main applications: fully-automatic MT, computer-assisted translation (CAT) and understandable rough translation[g]. This thesis approaches the two first applications from a statistical point of view.

Fully-automatic statistical MT consists in the development of statistical models that are automatically inferred from bilingual parallel texts. In this respect, there have been different proposals for statistical translation models ranging from word alignment translation models, such as the IBM models, HMM word alignment model, etc. to phrase-based and syntax-based translation models. These latter models are usually grounded on byproducts of the training process of word alignment models. However, none of these models directly addresses the common problem of context-specific translations in MT. This is the case of words whose meaning, and therefore their translation, depend on the domain or semantic context in which they are found. In this thesis, we introduce the idea of context-specific word alignment translation models by applying finite mixture modelling techniques to conventional word alignment translation models.

Nonetheless, current MT technology is still far from producing high quality translation without human intervention. This is the reason for developing CAT systems that can work in collaboration with translators to guarantee high quality translation, while easing and speeding up their work. The most popular instantiation of the CAT paradigm is implemented by post-editing tools based on translation memories. However, the lack of human-computer interactivity in a post-editing process prevents the MT system from adapting to the corrections of the human translator. Therefore, an interactive approach to CAT seems to be more adequate in this human-computer interaction setting. This thesis, partially developed in the framework of a European project devoted to the latter approach to CAT, presents how a fully-fledged MT system based on stochastic finite-state transducer (SFST) technology was integrated into an interactive and predictive CAT environment.

The objective of this thesis is to present new applications of existing technology in TC and CAT, and new models in TC and statistical MT based on the paradigm of finite mixture modelling. More precisely, the scientific contributions of this thesis can be divided into three groups as follows:

1. **Bilingual TC**. Bilingual TC is proposed as a new application in TC for which we suggest two general approaches. The first approach is the modelisation of each language independently, that in this thesis is instantiated in five mixture models based on the unigram model. These models were assessed on three tasks of increasing complexity. The second approach is a natural evolution of the latter. We derived a novel model that takes into account the word correlation across languages by combining the well-known M1 translation model with a unigram model. Comparative results with smoothed n-gram language models, support vectors machines and boosting techniques are also reported.

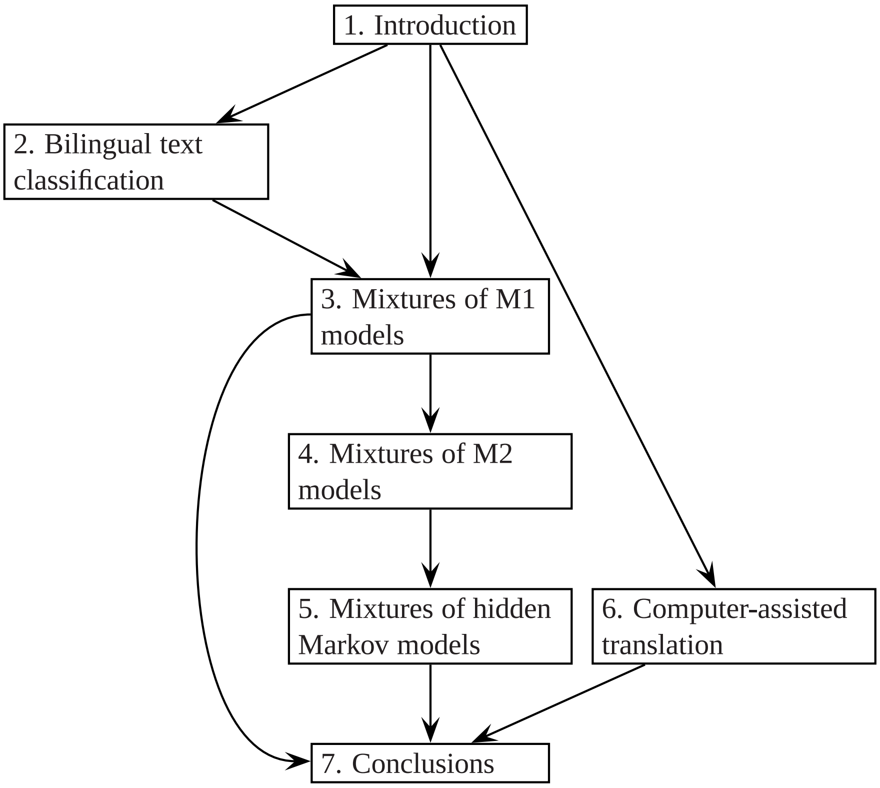
2. **Context-specific word alignment translation models**. Three translation

[g] Also known as gisting.

models, M1, M2 and HMM were extended to incorporate context information by means of finite mixture modelling. In the case of the M2 mixture model, we also derived a mixture extension of an iterative dynamic-programming search algorithm for the conventional M2 model that was evaluated in a small controlled task. Furthermore, an study of alignment and translation quality of these models is carried out on two shared tasks widely known in the statistical MT community.

3. **Interactive and predictive CAT based on SFST technology**. We adapt well-known error-correcting and n-best parsing algorithms in order to be integrated into a CAT system powered by SFST technology. This system was automatically and manually evaluated in the framework of a European project.

The above contributions are sequentially organised in 7 chapters that cover most of the work developed in this thesis. We recommend a sequential reading of the document should you wish to learn about the complete work, but that is not strictly necessary. If the reader is only interested in a specific research area, she can opt to read those related chapters taking into account the following dependency graph among chapters:


1. Introduction
2. Bilingual text classification
3. Mixtures of M1 models
4. Mixtures of M2 models
5. Mixtures of hidden Markov models
6. Computer-assisted translation
7. Conclusions

Five unigram models are proposed for bilingual TC along with its mixture extension and their experimental evaluation in Chapter 2. The M1 mixture model is introduced and applied to two different but related tasks, bilingual TC and statistical MT in Chapter 3. In bilingual TC, the M1 model is combined with a unigram model as a step forward in modeling bilingual texts, and evaluated on the two simpler tasks presented in Chapter 2. In statistical MT, the Viterbi alignments obtained as a byproduct of the M1 mixture model are directly evaluated on a reference alignment shared task for the statistical MT community and, used to train a statistical phrase-based system. This system is assessed in terms of translation quality on another large-scale shared task used as a reference to gauge the performance of state-of-art translation systems for European languages.

In Chapter 4, a mixture extension of the well-known M2 model and its dynamic-programming search algorithm are introduced and assessed on a semi-artificial task. This model is further evaluated on the two shared tasks previously mentioned. The HMM alignment model and its mixture version are also derived and assessed on the same shared tasks in Chapter 5. In Chapter 6, an interactive and predictive CAT system based on SFST technology is presented and automatically evaluated. In Chapter 7, a summary of the work and contributions presented in this thesis are discussed, followed by an outlook.

The appendix contains further details of the work presented in this thesis. In Appendix A, additional comparative experimental results using smoothed n-gram language models, support vector machines and boosting techniques in bilingual TC are reported. In Appendix B, a detailed derivation of some of the models presented in this thesis is introduced. Finally, a list of mathematical symbols and acronyms used throughout this thesis is presented in Appendix C.

Contents

Chapter 1
Preliminaries

1.1 Introduction

Natural language processing (NLP) is an hectic research field that aims at developing computer systems able to automatically generate and understand natural human language, both written and spoken. NLP is a subfield of artificial intelligence and linguistics, and as such tends to combine theories, methodologies and experts coming from both worlds. The challenging problems addressed by the NLP community sometimes require world knowledge to be successfully solved. In this thesis, we explore two important areas of NLP: text classification (TC) and machine translation (MT).

The purpose of TC is to convert an unstructured repository of documents into a structured one by automatically assigning documents to a predefined number of groups, in the case of text clustering, or to a set of predefined categories, in the case of text categorisation. Doing so, the task of storing, searching and browsing documents in these repositories is significantly simplified [Seb06]. Historically, the seminal article of Maron [Mar61] is taken as the starting point of TC. However it was not until the late eighties, early nineties when the need of organising large document collections increase the importance of TC. TC has been applied to news filtering, patent classification and more recently to web taxonomy and spam filtering. These days TC technology has reached a mature stage, nevertheless there are still open problems and challenges. See [Seb02] for an in-depth introduction to the evolution of TC over the last two decades.

MT is the use of computers to automate the translation of texts or utterances from one language into another language, while the underlying meaning remains the same. The history of MT goes back to the late forties with the famous publication of Weaver [Wea55], in which the problem of MT was tackled with cryptanalytic techniques inherited from the Second World War. This initial intensive research period was followed by a discreet and pragmatic epoch after the ALPAC report [BH60]. The seventies and eighties saw the proliferation of rule-based system such as Meteo [Tih82], Systran [Bil82] and METAL [BS85]. The contributions in the statistical MT field were minor until the early nineties, when the IBM group presented the Candide system [B$^+$94], a statistical translation system [B$^+$90, B$^+$93] that was demonstrated to be competitive with state-of-the-art systems. Since then, the development of statistical MT has experienced a major boost that seems to be reaching a technical plateau nowadays. See [HS92, JM00] for a detailed and thorough description of the history of MT.

In this chapter, first we briefly overview the state-of-the-art of TC and MT in Sections 1.2 and 1.3, respectively. Then, we focus on statistical MT in Section 1.4, and more precisely on statistical phrase-based models in Section 1.5. Next, we introduce in Section 1.6 some of the translation evaluation metrics that will be used throughout this thesis. Section 1.7 provides a short review of the state of the art in *computer-assisted translation* (CAT). While Section 1.8 is devoted to the well-known *expectation-maximisation* (EM) algorithm that is the parameter estimation algorithm of most of the models in this thesis. Following the EM algorithm in the previous section, we present the finite mixture modelling approach, which is the common factor in many models of this thesis, and its EM instantiation in Section 1.9. Finally, we summarise the scientific contributions of this thesis in Section 1.10.

1.2 Text classification

In the eighties, the most popular approach to TC was based on the development of rule-based systems with the help of knowledge engineers and domain experts. The main problem of this approach was the definition of hand-crafted rules and their maintenance. In the nineties, the rule-based approach was replaced by *pattern recognition* (PR) and *machine learning* (ML) approaches because their numerous advantages:

- The classifier is learnt from the observation of a set of preclassified documents by an inductive process.
- The same inductive process can be applied to generate different classifiers for different domains and applications. This fact introduces an important degree of automation in the construction of ad-hoc classifiers.
- The maintenance task is significantly simplified, since it only requires to retrain the classifier with the new working conditions.
- The existence of off-the-self software to train text classifiers requires less skilled man power than for constructing expert systems.
- The accuracy of text classifiers based on ML techniques competes with that of human experts and supersedes that of knowledge engineering methods.

In this thesis we focus on the statistical PR approach to TC. Under this approach the optimal decision rule assigning samples to classes is dictated by the principle of minimum *global risk* defined over the sample space. However, the global risk can be minimised by making as small as possible the risk for each sample x individually. In the case of classification tasks in which the evaluation metric is the *classification error rate* (CER), the risk of classifying a sample x into class c is the probability of error, i.e. the *posterior* probability of classifying x into a class that is not c. This latter probability is the sum over the posterior probability of all classes except for c. Therefore, in order to minimise the global risk, we must classify x into that class c that makes the sum of posterior probabilities over the rest of classes minimum. In other words, we must classify x into that class c that maximises the posterior probability $p(c \mid x)$. This is exactly what the Bayes decision (classification) rule says [DH73]

$$\hat{c}(x) = \arg\max_c p(c \mid x). \tag{1.1}$$

This posterior probability is usually decomposed according to the Bayes theorem

$$\hat{c}(x) = \arg\max_c \; \frac{p(c)\, p(x \mid c)}{p(x)} \tag{1.2}$$

where

$$p(x) = \sum_c p(c)\, p(x \mid c). \tag{1.3}$$

The term $p(x)$ is constant for all classes, so it is normally dropped and the usual form of Bayes decision rule arises

$$\hat{c}(x) = \arg\max_c \; p(c)\, p(x \mid c) \tag{1.4}$$

where $p(c)$ is the *prior* probability that is usually computed as the relative class frequency, and $p(x \mid c)$ is the *conditional* probability (density) function describing how likely is to observe x in class c.

As stated above, the Bayes rule is the optimal decision when we consider CER as evaluation metric. However, this is only the case under the assumption that we know the real probability distributions for $p(c)$ and $p(x \mid c)$. In practise, we can only compute approximations of these probability distributions.

In this thesis, the estimation of conditional probability distributions $p(x \mid c)$ for TC is based on smoothed n-gram language models [CG96, Jel97], and its mixture extension [KS93, IO99] in the case of the unigram language model.

Apart from those classifiers based on the statistical PR approach, different types of classifiers have been used in TC, including regression methods [FP94, IDLA95, LG94, SHP95], decision trees, neural networks [Mit96], incremental or batch methods for learning linear classifiers [SHP95, WPW95, DKR97, NGL97], classifier ensembles, including boosting methods [SS00], and support vector machines [Joa98]. While all these techniques still retain their popularity, it is fair to say that in recent years support vector machines and boosting have been the two dominant learning methods in TC. This fact is mainly due to their superiority on the Reuters task, which is the reference task in TC, however their performance is similar to that of other TC techniques in other tasks. The interested reader is referred to [Seb02] for an excellent review in TC.

1.2.1 Support vector machines and boosting techniques

In this section, we briefly review support vector machines (SVM) and boosting techniques since they will be used to obtain state-of-the-art comparative results in TC. We mostly provide a practical view of how these techniques were applied to multi-class classification tasks, as they are the focus of this thesis.

SVM were originally thought to be used as binary classifiers that define a hyperplane that maximises the margin between two classes, if the samples are linearly separable [Bur98, CST00]. Although there have been a generalisation of the 2-class problem [CS02], implemented in $SVM^{multiclass}$, an instance of SVM^{struct} [TJHA05], in practise binary classifiers based on the one-against-one approach, among others, seem to be the most adequate [HL02]. This simple yet effective approach consists in:

1. Define as many binary classifiers as possible class pairs.
2. Each binary classifier votes for a class.
3. Classify according to the majority voting criteria.

In this thesis, all the SVM experiments were carried out with the SVM^{light} toolkit [Joa99] adopting the approach to the multi-class problem commented above .

On the other hand, the idea behind boosting methods is to find a highly accurate classification rule by combining many weak or base hypotheses, each of which may be only moderately accurate. We assume access to a separate procedure called the weak learner or weak learning algorithm for computing the weak hypotheses. The boosting algorithm finds a set of weak hypotheses by calling the weak learner repeatedly in a series of rounds. These weak hypotheses are then combined into a single rule called the final or combined hypothesis. In the simplest version of the boosting method AdaBoost for single-label classification, the algorithm maintains a set of importance weights over training examples. These weights are used by the weak learning algorithm whose goal is to find a weak hypothesis with moderately low error with respect to these weights. Thus, the boosting algorithm can use these weights to force the weak learner to concentrate on the examples which are hardest to classify [SS00].

The implementation of the boosting algorithm employed in this thesis is BoosTexter [SS00]. In BoosTexter the weak learner is a one-level decision tree, and for our experiments the error, which we tried to minimise, will be measured in terms of *Hamming loss*[a]. In our case the input data will be text, therefore the condition

[a] There are other possible error functions such as ranking, in the sense of finding a hypothesis that places the correct labels at the top of the ranking.

that is checked at the root of the decision tree (weak learner) is the presence or absence of a given n-gram, for instance *Mahatma Gandhi*.

1.2.2 Machine-aided indexing

Most of the text classifiers reviewed so far assign a single class label to each document. However, in real-world TC applications, a document may receive more than one label reflecting the different topics included in the document. If this is the case, we would be facing a multi-label classification problem.

This multi-label characteristic is particularly common in *keyword assignment*, in which a list of descriptors (keywords or labels) from a thesaurus has to be assigned to a document. In this type of tasks, a classifier should first decide on the number of labels that will be assigned to a document, and then select the most suitable set of descriptors for that document. As the reader could devise, this task is significantly more complex than that of assigning a single label to each document.

In this setting the accuracy of text classifiers is usually far from being acceptable, and it is more convenient to look at our automatic text classifier as the backend of a machine-aided indexing (MAI) tool [Hod98, P$^+$03]. MAI tools usually assign to a document a list of keywords (descriptors) from a controlled vocabulary (thesaurus) for indexing purposes. This list of descriptors suggested by the system is reviewed by expert indexers to add and select those descriptors that are the most adequate.

The interest behind the development of indexing systems is not only the document classification capabilities *per se*, but also the possibility to access cross-lingual information [P$^+$03] through multilingual thesaurus, as EuroVoc [EC95], AgroVoc [FAO98], etc. In this scenario, documents in different languages are classified following the same multilingual thesaurus, and therefore, they use a common set of descriptor identifiers shared across languages for indexing purposes. Then, given a query (document), we could first identify the set of candidate descriptors for this query, and then retrieve those documents, no matter what their language is, labelled with these candidate descriptors.

In this case of multi-label TC, we adopted a simple training procedure that consists in learning a single-class classifier with all the documents with the same class label. While the classification rule in Eq. (1.1) is replaced by that providing a set of most probable class labels (descriptors)

$$\hat{S}_k(x) = \operatorname*{arg\,max}_{\substack{S \subset \Omega \\ |S|=k}} \min_{c \in S} p(c \,|\, x) \tag{1.5}$$

where Ω is the set of classes and $k \leq |\Omega|$.

Multi-label classifiers are usually evaluated in terms of precision and recall, or using a combination of these metrics like F-measure [BYRN99]. However, in the context of MAI tools is more meaningful to talk about macro-averaging precision and recall [Lew91] computed for a set of N documents

$$\overline{precision} = \frac{1}{N}\sum_{n}\frac{|S_n \cap R_n|}{|S_n|} \qquad \overline{recall} = \frac{1}{N}\sum_{n}\frac{|S_n \cap R_n|}{|R_n|} \tag{1.6}$$

where $S_n = \hat{S}_k(x_n)$, and R_n is the reference set of labels for the nth document.

The precision measure provides the ratio of correct labels over the total number of labels returned by the system, so it is an indicator of how precise is our system when providing a list of class labels. On the other hand, the recall measure reports the ratio of correct labels over the total number of reference labels, therefore it informs us about the coverage that the labels offered by the system provides of the set of reference labels. In this thesis, we will make use of these evaluation metrics as percentages when in Chapter 2 we report experimental results on a multi-label task.

1.3 Machine translation

In this section we review state-of-the-art applications and approaches in the field of MT. On the one hand, current MT technology is focused on three main applications:

- Fully-automatic MT in limited domains like weather forecast [LGLL05], hotel reception desk [ABC$^+$00], appointment scheduling, etc.
- Post-editing for CAT, understanding by post-editing the human amendment of automatic translations produced by an MT system.
- Understandable rough translation in which the aim is to allow a human to decide whether the translated text includes relevant information. For instance, this is used for document finding purposes or user assistance in software troubleshooting.

On the other hand, state-of-the-art MT approaches can be classified according to the level of analysis of the source sentence before translating:

- The interlingua approach consists in transforming the source sentence to a language independent semantic representation, the so-called interlingua, and translating that interlingua expression into the desired target language. The major drawback of this approach is its demanding knowledge resources to represent such language independent information. Further details of this approach can be found in [N$^+$92, NM92, A$^+$93].
- The transfer approach decomposes the translation process into three steps:

  Analysis. The source sentence is syntactically and semantically parsed to some abstract representation.

Transfer. A transformation from the source representation into the target representation is performed.

Generation. The final translation is generated from the target representation obtained in the previous step.

A review of transfer-based systems is presented in [HS92].

- The direct approach refers to the word-by-word translation from the source sentence into the target sentence. Under this approach we find example-based MT and statistical MT:

  Example-based MT. This approximation deals with the translation of new sentences by analysing, using different matching criteria, similar sentences previously translated. See [Som99] for a review of example-based MT[b].

  Statistical MT. A statistical model is inferred from translation examples and the translation process is derived from a statistical decision theory perspective. This thesis is mainly devoted to the statistical approximation to MT that will be further studied in the next section.

1.4 Statistical MT

The goal of MT is the automatic translation of a source sentence x into a target sentence y, being

$$\begin{aligned} x &= x_1 \ldots x_j \ldots x_{|x|} \quad && x_j \in \mathcal{X} \\ y &= y_1 \ldots y_i \ldots y_{|y|} \quad && y_i \in \mathcal{Y} \end{aligned}$$

where x_j and y_i denote source and target words, and $\mathcal{X}$ and $\mathcal{Y}$, the source and target vocabularies respectively.

In statistical MT, this translation process is usually presented as a decision process, where given a source sentence x, we will choose a target sentence $\hat{y}$ according to

$$\hat{y} = \arg\max_{y} \; p(y \,|\, x) \tag{1.7}$$

where $p(y \,|\, x)$ is the probability for y to be the actual translation of x or, in other words, the relative frequency of y being the actual translation of x. The so-called *search problem* is to compute a target sentence $\hat{y}$ for which this probability is maximum. Applying Bayes's theorem we can reformulate Eq. (1.7) as

$$\hat{y} = \arg\max_{y} \; p(x \,|\, y) \, p(y) \tag{1.8}$$

[b] Also known as memory-based MT [Bow02, Som03]

where the term $p(y \,|\, x)$ has been decomposed into a *translation model* $p(x \,|\, y)$ and a *language model* $p(y)$. Intuitively, the translation model is responsible for modelling the correlation between source and target sentence, but it can also be understood as a mapping function from target to source words. While the language model $p(y)$ represents the well-formedness of the candidate translation y. It should be noted that the term $p(x)$ has been intentionally omitted in the denominator of Eq. (1.8), since it is constant for a given x when maximising over y.

From a broader perspective, we can look at statistical MT as a specific instance of a classification problem where:

- The object to be classified is the sentence x to be translated.
- The set of possible classes are the set of possible sentences in the target language $y \in \mathcal{Y}^*$.
- The prior probability distribution is the language model $p(y)$.
- The conditional probability distribution is the translation model $p(x \,|\, y)$.

Therefore, under this point of view the decision rule stated in Eq. (1.7) is optimal under the assumption of a *zero-one* loss function. In statistical MT, the zero-one loss function is better known as *sentence error rate* (SER)[c] and considers that there is an error if the translation given by the system $\hat{y}$ is not identical to the reference translation.

In conclusion, by applying Eq. (1.7) we are minimising the probability of error using SER as a loss function. However, the SER measure provides a rough and superficial evaluation of the translation quality of a translation system and it is rarely used in favour of other more popular evaluation measures like *word error rate* (WER) and *bilingual evaluation understudy* (BLEU) [PRWZ01]. These evaluation measures, further explored in this thesis, suggest the usage of alternative loss functions, and therefore different decision rules that are closer to actual evaluation measures employed in statistical MT. An excellent discussion on the use of different loss functions in statistical MT can be found in [AF$^+$07].

A great variety of statistical translation models have been proposed since the IBM article was initially published [B$^+$90, B$^+$93]. In that article, the correspondence between source and target word positions is described by an alignment variable $a = a_1 \ldots a_j \ldots a_{|x|}$ where for each source position j, we have a target position $a_j \in \{0, \ldots, |y|\}$ to which is connected. The artificial zero position[d] is introduced to deal with source words with no direct mapping in the target sentence. The alignment variable is called a hidden variable since it is not directly observable in the translation process, but it naturally arises in the description of many probabilistic alignment models,

$$p(x \,|\, y) = \sum_{a \in \mathcal{A}(x,y)} p(x, a \,|\, y) \tag{1.9}$$

[c] SER in statistical MT is equivalent to CER in classification tasks.

[d] Better known in the literature as NULL or empty word.

where $\mathcal{A}(x, y)$ denotes the set of all possible alignments between x and y. The IBM article proposes five word alignment translation models of increasing complexity, that were implemented in the GIZA++ toolkit [ON03].

The M1 model, the first of the IBM models, is basically defined as a statistical bilingual dictionary, and it usually serves as a initialisation step for superior IBM models. Another interesting property of the M1 model is the concavity of its log-likelihood function, and therefore the uniqueness of a maximum value of this function under non-degenerated[e] initialisation. The M1 model has been widely employed in different applications of statistical MT, cross-lingual information retrieval and bilingual TC due to its simplicity and applicability of its parameter values.

In statistical MT, the M1 model has traditionally been an important ingredient in applications such as the alignment of bilingual sentences [Moo02], the alignment of syntactic tree fragments [DGP03], the segmentation of bilingual long sentences for improved word alignment [NCV03], the extraction of parallel sentences from comparable corpora [MFM04], the estimation of word-level confidence measures [UN07] and serves as inspiration for lexicalised phrase scoring in phrase-based systems [KOM03, Koe05]. Furthermore, it has also received attention to improve its nonstructural problems [Moo04].

Moreover, the M1 model has been recently applied to cross-lingual information retrieval with promising results [PJR07]. In that work, the authors use a training corpus made up by a set of query-relevant document pairs in a probabilistic cross-lingual information retrieval approach based on the M1 model.

In this thesis, the M1 model, as well as the M2 model, were studied and extended in Chapters 3 and 4, respectively. Apart from the IBM models, other word alignment translation models have been proposed, among them the homogeneous *hidden Markov alignment model* (HMM) [V$^+$96] that has received special attention [TIM02, DB05]. This thesis further elaborates upon the HMM model in Chapter 5.

The search problem presented in Eq. (1.8) was demonstrated to be an NP-complete problem [Kni99, UM06]. However various research groups have developed efficient search algorithms by using suitable simplifications and applying optimisation methods. Starting from the IBM work based on a stack-decoding algorithm [BPP96] over greedy [B$^+$94, WW98, G$^+$01] and integer-programming [G$^+$01] approaches to dynamic-programming search [GVC01, TN03]. This latter search approach was studied in Chapter 4.

Nevertheless, most of the current statistical MT systems pursue an alternative modelisation of the translation process different from that presented in Eq. (1.7). The posterior probability is modeled as a log-linear combination of feature func-

[e] Starting point in which none of the initial parameter values is zero.

tions [ON04] under the framework of maximum entropy [BPP96]

$$\hat{y} = \arg\max_{y} \sum_{m=1}^{M} \lambda_m h_m(x, y) \tag{1.10}$$

where λ_m is the interpolation weight and $h_m(x, y)$ is a function that assigns a score to the sentence pair (x, y).

Under this framework Eq. (1.8) can be seen as a special case where

$$h_1(x, y) = \log p(x \mid y) \tag{1.11}$$

$$h_2(x, y) = \log p(y) \tag{1.12}$$

and $\lambda_1 = \lambda_2 = 1$.

Most of state-of-the-art statistical MT systems are based on bilingual phrases [CB+07]. These bilingual phrases are sequences of words in the two languages and not necessarily phrases in the linguistic sense. The phrase-based approach to MT is further explored in Section 1.5.

Another approach which has become popular in recent years is grounded on the integration of syntactic knowledge into statistical MT systems [Wu96, YK01, GK04, Lin04, DP05]. This approach parses the sentence in one or both of the involved languages, defining then, the translation operations on parts of the parse tree. In [Chi07], Chiang constructs hierarchical transducers for translation. The model is a syntax-free grammar which is learnt from a bilingual corpus without any syntactic information. It consists of phrases which can contain sub-phrases, so that a hierarchical structure is induced.

The third main approach, which is currently investigated in statistical MT, is the modelling of the translation process as a finite-state transducer [ADB00, BR95, CV04, KN04, M+06]. This approach solves the translation problem by estimating a language model on sentences of extended symbols derived from the association of source and target words coming from the same bilingual pair. The translation transducer is basically an acceptor for this language of extended symbols. This approach is further explored in Chapter 6 where we introduce an interactive and predictive CAT system based on *stochastic finite-state transducers* (SFST).

1.5 Statistical phrase-based translation systems

1.5.1 Generative phrase-based models

In this section, we outline an example of generative phrase-based model [AFJC07] that will serve us to present the problems faced by this approach, and to motivate the introduction of heuristically estimated phrase-based systems in the next section.

Let (x, y) be a pair of source-target sentences, we introduce the conventional conditional probability $p(x \mid y)$ for the translation model. Let assume that x has been monotonically generated from T continuous segments that compose y. The resultant source and target non-empty segments are defined by $\mu = \{\mu_0, \mu_1, \ldots, \mu_T\}$ and $\gamma = \{\gamma_0, \gamma_1, \ldots, \gamma_T\}$ variables, respectively, where

$$\begin{aligned} 0 = &\mu_0 < \mu_1 < \ldots < \mu_T = |x| \\ 0 = &\gamma_0 < \gamma_1 < \ldots < \gamma_T = |y| \end{aligned} \tag{1.13}$$

so given two monotone, monolingual segmentations of x and y into T segments, μ and γ, their associated bilingual segmentation of x and y is defined as $s = s_1, s_2, \cdots, s_T$ with

$$s_t = (\mu_{t-1} + 1, \mu_t, \gamma_{t-1} + 1, \gamma_t) \qquad t = 1, \ldots, T. \tag{1.14}$$

An example of all possible bilingual segmentations for a source sentence of length 4 and a target sentence of length 5 is represented in Figure 1.1 as a direct multi-stage graph.

The initial stage of the graph has a single, artificial node labelled as “init”, which is only included to point to the initial segments of all the possible segmentations. There are 12 of such initial segments, vertically aligned on the first stage. Similarly, there are 15, 3 and 13 segments aligned on the second, third and final stages, respectively. The total number of segments is then 43. There is a unique segmentation of unit length, $s = (1415)$, which is represented by the rightmost path, but there are 12, 18 and 4 segmentations of length 2, 3 and 4, respectively; comprising 35 segmentations in total. As empty segments are not allowed, segmentation lengths range from one to the length of the shortest sentence.

Finally, our model for $p(x \,|\, y)$ can be seen as a full exploration of all possible bilingual segmentations of x and y,

$$p(x \,|\, y) = \sum_{T=1}^{\min(|x|,|y|)} \sum_{s} p(x, s, T \,|\, y) \tag{1.15}$$

where

$$p(x, s, T \,|\, y) = p(T \,|\, y)\, p(s \,|\, T, y)\, p(x \,|\, s, T, y) \tag{1.16}$$

is a generation process in which we first decide on the number of segments, then we select the sequence of segmentation states, and finally we generate the source segment.

The estimation of a phrase-based model as that presented above is a cumbersome problem that possess not only computational efficiency challenges, but also overwhelming data requirements. One of the main difficulties that phrase-based models have to cope with is the problem of the bilingual segmentation. In the model proposed above, this segmentation is explained by the hidden variables T and s which leads us to a large combinatorial number of possible segmentations to explore. As can be guessed, this problem is further aggravated with the length of the source and target sentence. Despite this obstacle, there have been several bold proposals for phrase-based models, from the joint probability model [MW02,

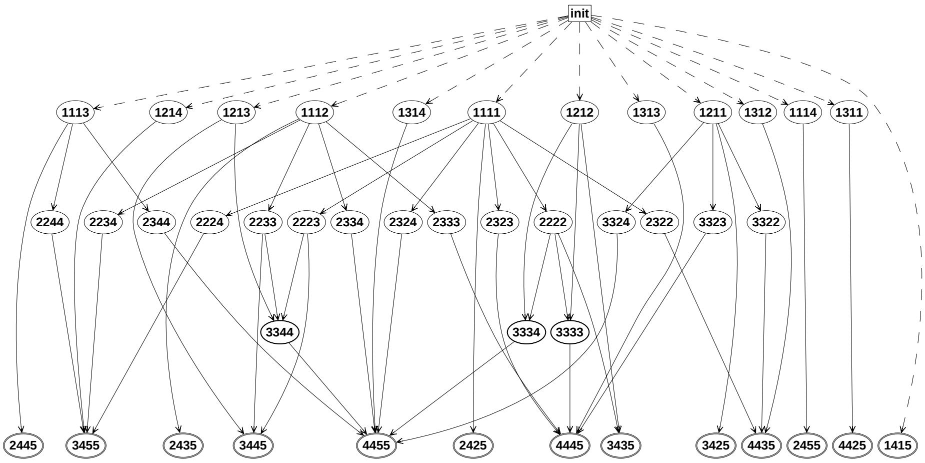


Figure 1.1: Directed, multi-stage graph representing all possible bilingual segmentations for a source sentence of length 4 and an target sentence of length 5. Each node defines a different segment; the first two digits of the node label are the segment limits in the source sentence, while the other two digits correspond to the target sentence.

BCBMOK06], over the HMM phrase-based models [DB05, AFJC07] to the statistical GIATI model [AFJCC08].

However, the most popular approach to the development of phrase-based systems has been the log-linear combination of heuristically estimated phrase-based models [KOM03, ON04], since these systems offer similar or even better performance than those based on generative phrase-based models [DGZK06].

1.5.2 Heuristic phrase-based models

The heuristic estimation of phrase-based models is grounded on the Viterbi alignments computed as a byproduct of word-based alignment models. The Viterbi alignment is defined as the most probable alignment given the source and target sentences and an estimation of the model parameters $\mathbf{\Theta}$,

$$\hat{a} = \arg\max_{a} p(a \mid x, y; \mathbf{\Theta}) \tag{1.17}$$

also rewritten

$$\hat{a} = \arg\max_{a} p(x, a \mid y; \mathbf{\Theta}). \tag{1.18}$$

The conventional alignments, for instance those provided by IBM models, disallow the connection of a source word with more than one target word. This unrealistic limitation negates the common linguistic phenomenon in which a word in one language is translated into more than one word in another language. To circumvent this problem, alignments are not only computed from the source language to the target language, but also from the target language to the source language. Doing so, we can reflect the fact that a single word is connected to more than one word.

Once the Viterbi alignments have been computed in both directions, there exist different heuristic algorithms to combine[f] them [KOM03, ON03]. These algorithms range from the intersection of both alignments in which we have high precision, but low recall alignments, to the union in which we have low precision, but high recall. In between, there are algorithms like the refined method [ON03] and the *grow-diag-final* [KOM03] that starting from the intersection, heuristically add additional alignment points taken from the union. The latter symmetrisation algorithm will be employed throughout this thesis to combine the Viterbi alignments provided by our word-based alignment translation models. This is a previous step, before extracting bilingual phrases, to construct a phrase-based system.

Bilingual phrase extraction is based on the concept of *consistency* of a bilingual phrase $(\overline{x}, \overline{y})$ (derived from a bilingual segmentation) with a word alignment a. Formally,

$$\begin{aligned} (\overline{x}, \overline{y}) \text{ consistent with } a \Leftrightarrow \quad & \forall x_j \in \overline{x} : (x_j, y_i) \in a \longrightarrow y_i \in \overline{y} \land \\ \land \quad & \forall y_i \in \overline{y} : (x_j, y_i) \in a \longrightarrow x_j \in \overline{x} \land \\ \land \quad & \exists x_j \in \overline{x},\, y_i \in \overline{y} : (x_j, y_i) \in a \end{aligned} \tag{1.19}$$

basically Eq. (1.19) means that a bilingual phrase is consistent if and only if, all the words in the source phrase are aligned to words in the target phrase, and there is at least one word in the source phrase aligned to a word in the target phrase.

Given the definition of consistency, all bilingual phrases (up to a maximum phrase length) that are consistent with the alignment resulting from the symmetrisation process are extracted.

The next step is to define functions that assign a score or a probability to a bilingual phrase in isolation or as part of a sequence of bilingual phrases in a given segmentation. These score functions are seamlessly integrated in a log-linear fashion under the maximum entropy framework.

The most commonly used score functions are the direct and inverse phrase translation probability estimated as a relative frequency

$$p_d(\overline{x} \,|\, \overline{y}) = \frac{count(\overline{x}, \overline{y})}{\sum\limits_{\overline{x}} count(\overline{x}, \overline{y})} \qquad p_i(\overline{y} \,|\, \overline{x}) = \frac{count(\overline{x}, \overline{y})}{\sum\limits_{\overline{y}} count(\overline{x}, \overline{y})} \tag{1.20}$$

as well as the direct and inverse lexical translation probability inspired in the M1 model [KOM03, CL07]. Other score functions are related to reordering capabilities, such as the distance-based reordering model [ON04] and the lexicalised reordering model [K$^+$05]. Additional score functions are phrase and word penalty to control the length of the translated sentence.

[f] This process is also known as symmetrisation.

The weight of each score function in the log-linear combination is adjusted on a development set with respect to a predefined criterion, usually BLEU. There are two popular techniques in statistical MT to carry out this process, minimum error rate training [Och03] and minimum Bayes risk [KB04]. The former criterion was used in this thesis to tune the weights of the log-linear model. Furthermore, the most common approach to the decoding process in log-linear models is the well-known multi-stack decoding algorithm [Koe04, ON04]. The Moses toolkit [K$^+$07], that implements an instantiation of this type of multi-stack decoding algorithms, will be used throughout this thesis to carry out most of the translation experiments. It should be noted that the Moses toolkit is employed at the user level to tune the weights of the score functions, and will allow us to indirectly evaluate the translation quality of the models proposed in this thesis.

1.6 Automatic MT evaluation metrics

In MT, the use of automatic evaluation metrics is imperative due to the high cost of human made evaluations. Also the need of rapid assessment of the translation quality of an MT system during its development and tuning phases is another reason for the usage of automatic metrics. These metrics are employed under the assumption that they correlate well with human judgements of translation quality. This arguable statement must be considered bearing in mind the low inter-annotator agreement on translation quality [CB$^+$07]. This fact makes automatic evaluation an open challenge in MT.

In this thesis, we mainly use two conventional translation evaluation metrics, WER and BLEU, although other measures like METEOR [BL05] and translation edit rate (TER) [S$^+$06] are becoming more and more popular.

The WER metric [A$^+$00, C$^+$04] is defined as the minimum number of word substitution, deletion and insertion operations required to convert the target sentence provided by the translation system into the reference translation, divided by the number of words of the reference translation. It can also be seen as the ratio of the edit distance between the system and the reference translation, and the number of words of the reference translation. This metric will allow us to compare our results to previous work on the same task. Even though the WER metric can value more than 1.0, it will be expressed as a percentage as it is commonly presented in the SMT literature. The WER metric can also be evaluated with respect to multiple references, however, in this thesis, we have a single reference translation at our disposal.

The BLEU score [PRWZ01] is the geometric mean of the modified[g] precision for different order of n-grams (usually from unigram up to 4-grams) between the target sentence and the reference translation, multiplied by an exponential brevity penalty (BP) factor that penalises those translations that are shorter than the ref-

[g] The number of occurrences of a word in a target sentence is limited to that of this word in the reference translation.

erence translation. Although some voices have been raised against BLEU as the dominant evaluation methodology over the past years [CBOK06], it is still a reference error measure for the evaluation of translation quality in MT systems. We take BLEU as a percentage ranging from 0.0 (worst score) to 100.0 (best score).

1.7 Computer-assisted translation

Present translation technology has not been able to deliver fully automated high-quality translations [NIS06, CB$^+$07]. Experts are almost unanimous about this [Isa96, Kay97a, Hut99, Arn03]. An alternative way to take advantage of the existing MT technologies is to use them in collaboration with human translators within a CAT framework.

Historically, CAT and MT have been considered as different but close technologies [Kay97b] and more so for one of the most popular CAT technologies, namely, translation memories. Translation memories are the basic ingredient of post-editing tools in which an MT system provides a complete translation using sentences previously translated, and a human expert corrects the possible errors incurred by the MT system. It should be noticed that there is no actual interaction between the MT system and the translator in this scenario, since they work as two isolated serial processes.

The main drawback of post-editing tools is that the serial process residing at the core of this technology, prevents the MT system from taking advantage of the knowledge of the human translator and the human translator cannot benefit from the adapting capability of the MT system. In contrast, an interactive approach to CAT [IC97] seems to be more adequate in this human-computer interactive setting. Interactivity in CAT has been explored for a long time. Systems have been designed to interact with human translators in order to solve different types of (lexical, syntactic or semantic) ambiguities [Slo85, WWC$^+$86]. Other interactive strategies have been considered for updating user dictionaries or for searching through dictionaries [Slo85, WWC$^+$86]. Specific proposals can be found in [Tom85, Zaj88, Y$^+$93, SZH97] among others.

An important contribution to interactive CAT technology was carried out around the TransType (TT) project [LFL00, LLL02, FLL02, Fos02]. This project entailed an interesting focus shift in which interaction directly aimed at the production of the target text, rather than at the disambiguation of the source text, as in former interactive systems. The idea proposed in that work was to embed data driven MT techniques within the interactive translation environment. The hope was to combine the best of both paradigms: CAT, in which the human translator ensures the high-quality output, and MT, in which the machine ensures a significant gain of productivity. Following these TT ideas, the innovative embedding proposed here consists in using a complete MT system to produce full target sentence hypotheses, or portions thereof, which can be partially or completely accepted and amended by a human translator. Each partial correct text segment is then used by the MT system as additional information to achieve further, hopefully improved suggestions.

A follow-up of the precursor TT project was TransType2 (TT2) project [Ato01]. In this project, a series of novel contributions were developed:

- Fully-fledged statistical MT systems produce complete sentence hypotheses as a response to user corrections, instead of single words suggestions as happened in the TT project.
- Systematic off-line experiments to simulate the specific conditions of interactive translation and analyse the results.
- Some of the systems developed in this project were successfully evaluated by professional translators under real working conditions [MNS05, Mac06].

This thesis presents one of the three CAT systems developed in the framework of the TT2 project, as said before, this system based on SFST technology is introduced in Chapter 6.

In the CAT scenario, the speech was proposed as an alternative means of interaction [VCR$^+$06] and as a tool to dictate translations [KZN05, KZN06].

1.8 The expectation-maximisation algorithm

In this thesis, we present a series of probabilistic models $p(x; \mathbf{\Theta})$ that are governed by their corresponding set of parameters $\mathbf{\Theta}$. Supposing that we have N samples that have been randomly drawn from $p(x; \mathbf{\Theta})$

$$X = \{x_1, \ldots, x_n, \ldots, x_N\},$$

we can compute the log-likelihood[h] of $\mathbf{\Theta}$ w.r.t. N independent samples as

$$L(\mathbf{\Theta}; X) = \log p(X; \mathbf{\Theta}) = \sum_{n=1}^{N} \log p(x_n; \mathbf{\Theta}). \tag{1.21}$$

The log-likelihood can be thought of as a function of the parameters $\mathbf{\Theta}$ where the data X is fixed. Our goal will be to find an estimation $\hat{\mathbf{\Theta}}$ that maximises the log-likelihood of $\mathbf{\Theta}$,

$$\hat{\mathbf{\Theta}} = \arg\max_{\mathbf{\Theta}} L(\mathbf{\Theta}; X). \tag{1.22}$$

Depending on the form of $p(x; \mathbf{\Theta})$ this maximisation can be easy or hard. For example, if $p(x; \mathbf{\Theta})$ is a D-nomial distribution of parameters $x_+!$ and

$$\mathbf{\Theta} = (p_1, \ldots, p_d, \ldots, p_D) \tag{1.23}$$

[h] We take the logarithm of the likelihood because it is analytically and computationally easier to work with.

that is,

$$p(x;\mathbf{\Theta}) = \frac{x_{+}!}{\prod_{d=1}^{D} x_d!} \prod_{d=1}^{D} p_d^{x_d} \tag{1.24}$$

then, the maximum likelihood estimate of $\mathbf{\Theta}$ w.r.t. N independent D-nomial samples can be easily computed by taking partial derivatives of Eq. (1.21) w.r.t. p_d and equating to zero, subject to the constraint that $\mathbf{\Theta}$ defines a p.f. Doing so, we have that $\hat{\mathbf{\Theta}}$ is

$$p_d = \frac{\sum\limits_{n=1}^{N} x_{nd}}{\sum\limits_{d=1}^{D}\sum\limits_{n=1}^{N} x_{nd}} \tag{1.25}$$

where x_{nd} is the number of occurrences of the d-th event in the n-th sample. However, this is much more complicated for many interesting models, including nearly all of those studied in this thesis.

Fortunately, maximum likelihood estimation of all "complicated" models studied in this work can be reliably accomplished by the EM algorithm [DLR77, Wu83]. The EM algorithm considers X to be incomplete data which can be *completed* by addition of missing (*hidden* or *latent*) data Z. This results in a many-to-one mapping from the complete to the incomplete models,

$$p(X;\mathbf{\Theta}) = \sum_{Z\in\mathcal{Z}} p(X,Z;\mathbf{\Theta}). \tag{1.26}$$

where $\mathcal{Z}$ is the domain from which Z takes value. The marginalisation in Eq. (1.26), represented as the sum over the domain of the hidden variable Z, is the case of all the models presented in this thesis, since they only involve discrete variables. However, this sum is replaced by an integral in the case of continuous variables, or a combination of sum and integral when the marginalisation is carried out over discrete and continuous variables.

The EM algorithm works iteratively in two basic steps. Firstly, the E step computes the expected value of the logarithm of $p(X,Z;\mathbf{\Theta})$ w.r.t. the posterior $p(Z \mid X;\mathbf{\Theta}^{(k)})$,

$$Q(\mathbf{\Theta} \mid \mathbf{\Theta}^{(k)}) = E(\log p(X,Z;\mathbf{\Theta}) \mid X, \mathbf{\Theta}^{(k)}). \tag{1.27}$$

Secondly, the M step maximises $Q(\mathbf{\Theta} \mid \mathbf{\Theta}^{(k)})$ to obtain a new estimation of $\mathbf{\Theta}$,

$$\mathbf{\Theta}^{(k+1)} = \arg\max_{\mathbf{\Theta}} Q(\mathbf{\Theta} \mid \mathbf{\Theta}^{(k)}). \tag{1.28}$$

These two steps are repeated for a number of iterations or until convergence[i]. There exists a modified version of the algorithm known as the generalised EM [DLR77],

[i] Normally, the condition for convergence is a relative increase of log-likelihood from iteration k to $k+1$ below a given threshold.

in which the M step is only required to fulfil the condition $Q(\mathbf{\Theta}^{(k+1)} \,|\, \mathbf{\Theta}^{(k)}) > Q(\mathbf{\Theta}^{(k)} \,|\, \mathbf{\Theta}^{(k)})$. In any case, the algorithm converges to a local maximum of the likelihood function.

Let us now consider the conventional case in which the missing data consists of N independent and identically distributed hidden variables[j],

$$Z = \{z_1, \ldots, z_n, \ldots, z_N\} \tag{1.29}$$

and thus, Eq. (1.26) factorises over the joint distribution,

$$\begin{aligned} p(X; \mathbf{\Theta}) &= \sum_{Z \in \mathcal{Z}} \prod_{n=1}^{N} p(x_n, z_n; \mathbf{\Theta}) \\ &= \sum_{z_1} \cdots \sum_{z_N} \prod_{n=1}^{N} p(x_n, z_n; \mathbf{\Theta}) \\ &= \sum_{z_1} p(x_1, z_1; \mathbf{\Theta}) \left[\sum_{z_2} \cdots \sum_{z_N} \prod_{n=2}^{N} p(x_n, z_n; \mathbf{\Theta}) \right] \\ &= \prod_{n=1}^{N} \sum_{z_n} p(x_n, z_n; \mathbf{\Theta}). \end{aligned} \tag{1.30}$$

Similarly, the joint probability for N independent samples becomes

$$p(X, Z; \mathbf{\Theta}) = \prod_{n=1}^{N} p(x_n, z_n; \mathbf{\Theta}). \tag{1.31}$$

Thus, the E step can be rewritten as

$$Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) = \sum_{n=1}^{N} E(\log p(x_n, z_n; \mathbf{\Theta}) \,|\, x_n, \mathbf{\Theta}^{(k)}). \tag{1.32}$$

where we compute the expected value of the missing data z_n for each sample independently.

1.9 Finite mixture modelling

Finite mixture modelling is a popular approach for the estimation of probability (density) functions in PR [TSM85, JDM00]. Mixtures are flexible enough for finding an appropriate tradeoff between model complexity and the amount of training data available. Usually, model complexity is controlled by varying the number of mixture components while keeping the same (often simple) parametric form for all components. Although most research on mixture models has concentrated on

[j] Each hidden variable z_n completes its corresponding incomplete (observed) data sample x_n.

mixtures for continuous data, there are many tasks for which discrete mixtures are better suited, for instance those related to model natural language.

A T-component mixture model is a probability (density) function of the form

$$p(x;\mathbf{\Theta}) = \sum_{t=1}^{T} p(t)\, p(x \mid t; \mathbf{\Theta}_t) \tag{1.33}$$

where for each component t, $p(t)$ is its mixture prior or coefficient and $p(x \mid t; \mathbf{\Theta}_t)$ is its component-conditional probability (density) function governed by the component parameter vector $\mathbf{\Theta}_t$. It can be seen as a generative model that first selects the tth component (or topic) with probability $p(t)$ and then generates x in accordance with $p(x \mid t; \mathbf{\Theta}_t)$. Thus, the global vector of parameters $\mathbf{\Theta}$ is

$$\mathbf{\Theta} = (p(1), \dots, p(t), \dots, p(T); \mathbf{\Theta}')^t \tag{1.34}$$

where

$$\mathbf{\Theta}' = (\mathbf{\Theta}_1, \dots, \mathbf{\Theta}_t, \dots, \mathbf{\Theta}_T)^t. \tag{1.35}$$

We can provide an equivalent representation of the finite mixture model presented above using an indicator vector $\boldsymbol{z}$,

$$\boldsymbol{z} = (z_1, \dots, z_t, \dots, z_T)^t \tag{1.36}$$

with 1 in the position corresponding to the tth component generating x, and zeros elsewhere

$$\boldsymbol{z} = (0, \dots, 0, \underset{\substack{\Uparrow \\ t}}{1}, 0, \dots, 0)^t. \tag{1.37}$$

Therefore, the domain of the hidden variable $\boldsymbol{z}$ is composed of vectors with 1 in a single position and zeros elsewhere.

This indicator vector replaces the integer latent variable t introduced in Eq. (1.33), simplifying the derivation of the EM algorithm, while being equivalent to the integer representation.

Then, the mixture model in Eq. (1.33) is rewritten in terms of an indicator vector as

$$\begin{aligned} p(x;\mathbf{\Theta}) &= \sum_{\boldsymbol{z}} p(x, \boldsymbol{z}; \mathbf{\Theta}) \\ &= \sum_{\boldsymbol{z}} p(\boldsymbol{z})\, p(x \mid \boldsymbol{z}; \mathbf{\Theta}') \end{aligned} \tag{1.38}$$

where $p(\boldsymbol{z})$ is a multinomial p.f.,

$$p(\boldsymbol{z}) = \prod_{t=1}^{T} p(t)^{z_t} \tag{1.39}$$

being $p(\boldsymbol{z}) = p(t)$ when $\boldsymbol{z}$ values 1 in the tth position and zeros elsewhere, and $p(x \,|\, \boldsymbol{z})$ is a component-conditional p.f. over x,

$$p(x \,|\, \boldsymbol{z}; \boldsymbol{\Theta}') = \prod_{t=1}^{T} p(x \,|\, t; \boldsymbol{\Theta}_t)^{z_t}. \tag{1.40}$$

being $p(x \,|\, \boldsymbol{z}; \boldsymbol{\Theta}') = p(x \,|\, t; \boldsymbol{\Theta}_t)$ when $\boldsymbol{z}$ values 1 in the tth position and zeros elsewhere. Thus, the general form for a finite mixture model becomes

$$p(x; \boldsymbol{\Theta}) = \sum_{\boldsymbol{z}} \prod_{t=1}^{T} \left[p(t)\, p(x \,|\, t; \boldsymbol{\Theta}_t) \right]^{z_t}. \tag{1.41}$$

which is equivalent to Eq. (1.33).

Now, let X be a set of samples available to learn $\boldsymbol{\Theta}$ that governs a finite mixture model as described above. This is a statistical parameter estimation problem since the mixture is a p.f. of known functional form, and all that is unknown is a parameter vector including the priors $p(t)$ and component-conditional parameters in $\boldsymbol{\Theta}_t$. The optimal parameter values maximise the log-likelihood function of $\boldsymbol{\Theta}$ w.r.t. X,

$$L(\boldsymbol{\Theta}; X) = \sum_{n=1}^{N} \log \sum_{\boldsymbol{z}_n} \prod_{t} \left[p(t)\, p(x_n \,|\, t; \boldsymbol{\Theta}_t) \right]^{z_{nt}}. \tag{1.42}$$

that can be estimated with an instantiation of the EM algorithm presented in Section 1.8.

The EM algorithm for a finite mixture model entails the application of the E step in Eq. (1.32). Here, we consider the indicator vector $\boldsymbol{z}$ to be the missing data z_n drawn from Z in Eq. (1.29), but redefined as

$$Z = \{ \boldsymbol{z}_1, \ldots, \boldsymbol{z}_n, \ldots, \boldsymbol{z}_N \}. \tag{1.43}$$

So, the E-step computes the expected value of the logarithm of $p(x_n, \boldsymbol{z}_n; \boldsymbol{\Theta})$ given the observed data x_n and the current estimation of $\boldsymbol{\Theta}$, $\boldsymbol{\Theta}^{(k)}$

$$\begin{aligned} Q(\boldsymbol{\Theta} \,|\, \boldsymbol{\Theta}^{(k)}) &= \sum_{n=1}^{N} E(\log p(x_n, \boldsymbol{z}_n; \boldsymbol{\Theta}) \,|\, x_n, \boldsymbol{\Theta}^{(k)}) \\ &= \sum_{n=1}^{N} \sum_{t=1}^{T} z_{nt}^{(k)} \left[\log p(t) + \log p(x_n \,|\, t; \boldsymbol{\Theta}_t) \right]. \end{aligned} \tag{1.44}$$

where $z_{nt}^{(k)}$ is the posterior probability of x_n being generated from the tth component,

$$z_{nt}^{(k)} = \frac{p(t)\, p(x_n \,|\, t; \boldsymbol{\Theta}_t^{(k)})}{\sum_{t'=1}^{T} p(t')\, p(x_n \,|\, t'; \boldsymbol{\Theta}_{t'}^{(k)})}. \tag{1.45}$$

The M-step maximises the function Q in Eq. (1.44) subject to the constraint that mixture coefficients must sum up to one, along with additional constraints imposed by the normalisation of the parameters defined in $\boldsymbol{\Theta}_t$. These constraints are incorporated into the maximisation problem in Eq. (1.28) via Lagrange multipliers,

$$\boldsymbol{\Theta}^{(k+1)} = \arg\max_{\boldsymbol{\Theta}} \max_{\lambda} \; Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)}) + \Lambda \tag{1.46}$$

with

$$\Lambda = -\lambda \left(\sum_{t=1}^{T} p(t) - 1 \right) . \tag{1.47}$$

where λ is the Lagrange multiplier to normalise the mixture coefficients. Additional Lagrange multipliers are usually needed to enforce the normalisation constraints of the parameters defined in $\boldsymbol{\Theta}_t$.

In the case of mixture coefficients, we take derivatives of $Q + \Lambda$ w.r.t. $p(t)$ and λ and equate them to zero in order to obtain their corresponding update equation,

$$p(t)^{(k+1)} = \frac{1}{N} \sum_{n=1}^{N} z_{nt}^{(k)} \qquad \forall t \tag{1.48}$$

that can be understood as the average contribution (responsibility) of the tth component to generate the training set, or alternatively, as the relative count of training samples drawn from the tth component. Similarly, the derivative of Q w.r.t. $\boldsymbol{\Theta}_t$ and its Lagrange multipliers (if any) must be equal to zero, so as to obtain the corresponding update equations for $\boldsymbol{\Theta}_t^{(k)}$.

In this thesis, we exclude the problem of the estimation of the optimal number of components per mixture. Instead, we prefer to study the evolution of the evaluation metrics as a function of the number of components for the different tasks in bilingual TC and statistical MT. This study is constrained by memory requirements.

1.10 Scientific contributions

The objective of this thesis is to present new applications of existing technology in TC and CAT, as well as new models in TC and statistical MT based on the paradigm of mixture modelling. More precisely, the scientific contributions of this thesis are:

1. **Bilingual TC**. Bilingual TC is proposed as a new application in TC along with novel models that are capable to deal with bilingual information and learn word correlation across languages. Five models based on the unigram model and its corresponding mixture extension are presented. These models were tested on three tasks of increasing complexity going from a semi-artificial task, over a real small task, to a real complex task. The two

first tasks were assessed in terms of CER, while the latter task was evaluated on precision-recall figures due to its multilabel nature and the working conditions from which it is extracted. As a natural evolution of the five unigram models previously mentioned, we derived a novel model that combines a unigram distribution with the well-known M1 translation model. The unigram-M1 model incorporates word correlation constraints into the bilingual classification model in order to fully exploit the bilingual information. This model was assessed on the first two proposed tasks. Along with these models, smoothed n-gram language models, support vectors machines and boosting techniques are included in the evaluation for comparative purposes and for the study of variable-size word-context information in the framework of bilingual TC.

2. **Context-specific word alignment translation models**. Three fundamental word alignment statistical translation models, M1, M2 and HMM, were extended to incorporate context information by applying mixture modelling. The word alignment translation models proposed so far in the literature ignore the context information from which bilingual sentences are extracted, however it is unanimously accepted that a word has different meanings depending on the semantic domain in which is found. The mixture models presented in this thesis aim at capturing domain information by developing context-specific word alignment translation models without supervision. The M1, M2 and HMM mixture models proposed were evaluated from two points of view on real shared tasks. On the one hand, we assessed the quality of the Viterbi alignments generated by these models and on the other hand, we indirectly measure the translation quality of these models by feeding its Viterbi alignments into a statistical phrase-based system. In the case of the M2 mixture model, we also developed a mixture extension of an iterative dynamic-programming search algorithm for the conventional M2 model. This search algorithm allows us to directly gauge the translation quality of the M2 mixture model on a semi-artificial translation task.

3. **Interactive and predictive CAT based on SFST technology**. Current MT technology is still far from producing fully-automatic high quality translation. Alternatively, CAT systems seek to integrate MT techniques in the human translator activity in order to increase their productivity and guarantee high quality translations. Following this idea, our proposal is to perform such a human-computer synergy mediating the target sentence that is being translated, that is, the translator guides the translation process by correcting the suggestions that the system offers. More precisely, in our case the backend MT technology for such an interactive and predictive CAT system is powered by SFSTs. The capability of this formalism to provide adequate translations and the existence of efficient parsing algorithms justify its selection given the tight usability and real-time constraints that these interactive sys-

tems require. The work carried out in this part of the thesis is focused on the adaptation, development and integration of existing algorithms in the field of finite-state machines to construct an interactive and predictive CAT system. The resultant system is automatically evaluated in terms of off-line translation quality and on-line typing effort reduction in two corpora of different complexity. Furthermore, this system was manually assessed by translators in controlled user-trial session. This latter work was externally developed in the framework of a European project.

BIBLIOGRAPHY

[A$^+$93] D.J. Arnold et al. *Machine Translation: an Introductory Guide*. Blackwells-NCC, London, 1993.

[A$^+$00] J. C. Amengual et al. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103, 2000.

[ABC$^+$00] J.C. Amengual, J.M. Benedí, F. Casacuberta, A. Castaño, A. Castellanos, V. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, and J.M. Vilar. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103, 2000.

[ADB00] H. Alshawi, S. Douglas, and S. Bangalore. Learning dependency translation models as collections of finite-state head transducers. *Computational Linguistics*, 26(1):45–60, 2000.

[AF$^+$07] J. Andrés-Ferrer et al. On the use of different loss functions in statistical pattern recognition applied to machine translation. *To appear in Pattern Recognition Letters*, 2007.

[AFJC07] J. Andrés-Ferrer and A. Juan-Císcar. A phrase-based hidden markov model approach to machine translation. In *Proceedings of New Approaches to Machine Translation*, pages 57–62, January 2007.

[AFJCC08] J. Andrés-Ferrer, A. Juan-Císcar, and F. Casacuberta. Statistical estimation of rational transducers applied to machine translation. *Applied Artificial Intelligence*, page In press, 2008.

[Arn03] D. J. Arnold. *Computers and Translation: A translator's guide*, chapter 8, pages 119–142. John Benjamins, Amsterdam, 2003.

[Ato01] Atos Origin, Instituto Tecnológico de Informática, RWTH Aachen, RALI Laboratory, Celer Soluciones and Société Gamma and Xerox Research Centre Europe. TransType2 - Computer Assisted Translation. Project Technical Annex., 2001.

[B$^+$90] P. F. Brown et al. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

[B$^+$93] P. F. Brown et al. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[B+94] A.L. Berger et al. The candide system for machine translation. In *Proc. of HLT'94*, pages 157–162, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

[BCBMOK06] A. Birch, C. Callison-Burch, Miles M. Osborne, and P. Koehn. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 154–157, New York City, New York, USA, June 2006. Association for Computational Linguistics.

[BH60] Y. Bar-Hillel. The present status of automatic translation of languages. *Advances in Computers*, 1:91–163, 1960.

[Bil82] R. Billmeyer. Zu den linguistischen Grundlagen von SYSTRAN. *Multilingua*, 2(1):83–96, 1982.

[BL05] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, USA, June 2005. Association for Computational Linguistics.

[Bow02] L. Bowker. *Computer-aided translation technology: A practical introduction*, chapter 5: Translation-memory systems, pages 92–127. Didactics of Translation. University of Ottawa Press, 2002.

[BPP96] A. L. Berger, V. J. Della Pietra, and S. A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[BR95] S. Bangalore and G. Riccardi. A finite-state approach to machine translation. In *Proc. of NAACL'01*, pages 1–8, Morristown, NJ, USA, June 1995. Association for Computational Linguistics.

[BS85] W. Bennett and J. Slocum. The RLC machine translation system. *Computational Linguistics*, 1:91–163, 1985.

[Bur98] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[BYRN99] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[C+04] F. Casacuberta et al. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech and Language*, 18:25–47, 2004.

[CB+07] C. Callison-Burch et al. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[CBOK06] C. Callison-Burch, M. Osborne, and P. Koehn. Re-evaluating the role of bleu in machine translation research. In *Proc. of ACL'06*, pages 249–256, Trento, Italy, April 2006. Association for Computational Linguistics.

[CG96] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL'96*, pages 310–318, Morristown, NJ, USA, June 1996. Association for Computational Linguistics.

[Chi07] D. Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007.

[CL07] T. Cohn and M. Lapata. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proc. of ACL'07*, pages 728–735, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[CS02] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal Machine Learning Research*, 2:265–292, 2002.

[CST00] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000.

[CV04] F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

[DB05] Y. Deng and W. Byrne. HMM word and phrase alignment for statistical machine translation. In *Proc. of HLT-EMNLP'05*, pages 169–176. Association for Computational Linguistics, October 2005.

[DGP03] Y. Ding, D. Gildea, and M. Palmer. An algorithm for word-level alignment of parallel dependency trees. In *Proc. of MT Summit IX*, pages 95–101, September 2003.

[DGZK06] J. DeNero, D. Gillick, J. Zhang, and D. Klein. Why generative phrase models underperform surface heuristics. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 31–38, New York City, June 2006. Association for Computational Linguistics.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[DKR97] I. Dagan, Y. Karov, and D. Roth. Mistakedriven learning in text categorization. In *Proc. of EMNLP'97*, pages 55–63, Cambridge, MA, USA, 1997.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[DP05] Y. Ding and M. Palmer. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proc. of ACL'05*, pages 541–548, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[EC95] EC. Thesaurus eurovoc - volume 2: Subject-oriented version. Annex to the index of the Official Journal of the EC, Office for Official Publications of the EC, 1995. http://europa.eu.int/celex/eurovoc.

[FAO98] FAO. Multilingual agricultural thesaurus. World Agricultural Information Center, 1998. http://www.fao.org/scripts/agrovoc/frame.htm.

[FLL02] G. Foster, P. Langlais, and G. Lapalme. User-friendly text prediction for translators. In *Proc. of EMNLP'02*, pages 148–155, Morristown, NJ, USA, July 2002. Association for Computational Linguistics.

[Fos02] G. Foster. *Text Prediction for Translators*. PhD thesis, Université de Montréal, May 2002.

[FP94] N. Fuhr and U. Pfeifer. Probabilistic information retrieval as combination of abstraction, inductive learning and probabilistic assumptions. *ACM Transactions on Information Systems*, 12(1):92–115, 1994.

[G$^+$01] U. Germann et al. Fast decoding and optimal decoding for machine translation. In *Proc. of ACL'01*, pages 228–235, Morristown, NJ, USA, June 2001. Association for Computational Linguistics.

[GK04] J. Graehl and K. Knight. Training tree transducers. In *Proc. of HLT-NAACL'04*, pages 105–112, Morristown, NJ, USA, May 2004. Association for Computational Linguistics.

[GVC01] I. García-Varea and F. Casacuberta. Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *Proc. of MT Summit VIII*, pages 115–120, Santiago de Compostela, Spain, 2001.

[HL02] Ch-W. Hsu and Ch-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[Hod98] G. Hodge. CENDI agency indexing system descriptors: A Baseline Report. Technical report, Information International Associates, Inc., 1998.

[HS92] J. Hutchins and H. L. Somers. *An introduction to machine translation*. Academic Press, 1992.

[Hut99] J. Hutchins. Retrospect and prospect in computer-based translation. In *Proc. of MT Summit VII*, pages 30–44, 1999.

[IC97] P. Isabelle and K. Church. Special issue on new tools for human translators. *Machine Translation*, 12(1–2), 1997.

[IDLA95] D. J. Ittner, D. D. D. Lewis, and D. D. Ahn. Text categorization of low quality images. In *Proc. of SDAIR'95*, pages 301–315, April 1995.

[IO99] R. M. Iyer and M. Ostendorf. Modelling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech & Audio Processing*, 7(1):30–39, 1999.

[Isa96] P. Isabelle. The state of machine translation in 1996. Technical report, Invited report prepared for the National Research Council of the United States of America, 1996.

[JDM00] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Trans. on PAMI*, 22(1):4–37, 2000.

[Jel97] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

[JM00] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

[Joa98] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proc. of ECML'98*, pages 137–142, April 1998.

[Joa99] T. Joachims. Making large-scale support vector machine learning practical. *Advances in kernel methods: support vector learning*, pages 169–184, 1999.

[K$^+$05] P. Koehn et al. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *Proc. of IWSLT'05*, October 2005.

[K$^+$07] P. Koehn et al. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL'07: Demo and Poster Sessions*, pages 177–180, Morristown, NJ, USA, June 2007. Association for Computational Linguistics.

[Kay97a] M. Kay. It's still the proper place. *Machine Translation*, 12(1-2):35–38, 1997.

[Kay97b] M. Kay. The proper place of men and machines in language translation. *Machine Translation*, 12:3–23, 1997.

[KB04] S. Kumar and W. J. Byrne. Minimum bayes-risk decoding for statistical machine translation. In *Proc. of HLT-NAACL'04*, pages 169–176, Morristown, NJ, USA, May 2004. Association for Computational Linguistics.

[KN04] S. Kanthak and H. Ney. FSA: an efficient and flexible C++ toolkit for finite state automata using on-demand computation. In *Proc. of ACL'04*, page 510, Morristown, NJ, USA, July 2004. Association for Computational Linguistics.

[Kni99] K. Knight. Decoding complexity in word-replacement translation models. *Computional Linguistics*, 25(4):607–615, 1999.

[Koe04] P. Koehn. Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *Proceedings of AMTA'04*, pages 115–124, Washington, District of Columbia, USA, September-October 2004.

[Koe05] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proc. of the MT Summit X*, pages 79–86, September 2005.

[KOM03] P. Koehn, F.J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. of NAACL'03*, pages 48–54, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[KS93] R. Kneser and V. Steinbiss. On the dynamic adaptation of stochastic language models. In *Proc. of ICASSP'93*, volume II, pages 586–589, April 1993.

[KZN05] S. Khadivi, A. Zolnay, and H. Ney. Automatic text dictation in computer-assited translation. In *Proc. of Interspeech'05*, pages 2265–2268, Lisbon, Portugal, September 2005.

[KZN06] S. Khadivi, R. Zens, and H. Ney. Integration of speech to computer-assisted translation using finite-state automata. In *Proc. of the COLING/ACL'06*, pages 467–474, Sydney, Australia, July 2006. Association for Computational Linguistics.

[Lew91] D. D. Lewis. Evaluating text categorization. In *Proc. of HLT'91: Workshop of Speech and Natural Language Workshop*, pages 312–318, Morristown, NJ, USA, February 1991. Morgan Kaufmann / Association for Computational Linguistics.

[LFL00] P. Langlais, G. Foster, and G. Lapalme. Unit completion for a computer-aided translation typing system. *Machine Translation*, 15(4):267–294, 2000.

[LG94] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In W. Bruce Croft and Cornelis J. Van Rijsbergen, editors, *Proc. of SIGIR'94*, pages 3–12, Dublin, Ireland, 1994. Springer Verlag, Heidelberg, Germany.

[LGLL05] Philippe Langlais, Simona Gandrabur, Thomas Leplus, and Guy Lapalme. The long-term forecast for weather bulletin translation. *Machine Translation*, 19(1):83–112, March 2005.

[Lin04] D. Lin. A path-based transfer model for machine translation. In *Proc. of COLING'04*, page 625, Morristown, NJ, USA, August 2004. Association for Computational Linguistics.

[LLL02] P. Langlais, G. Lapalme, and M. Loranger. Transtype: Development-evaluation cycles to boost translator's productivity. *Machine Translation*, 15(4):77–98, 2002.

[M$^+$06] J. B. Mariño et al. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549, 2006.

[Mac06] E. Macklovitch. TransType2: The last word. In *Proceedings of the 5th International Conference on Languages Resources and Evaluation (LREC 2006)*, pages 167–172, Genoa, Italy, May 2006.

[Mar61] M. E. Maron. Automatic indexing: An experimental inquiry. *J. ACM*, 8(3):404–417, 1961.

[MFM04] D.S. Munteanu, A. Fraser, and D. Marcu. Improved machine translation performance via parallel sentence extraction from comparable corpora. In *Proc. of HLT-NAACL'04*, pages 265–272, Morristown, NJ, USA, May 2004. Association for Computational Linguistics.

[Mit96] T. M. Mitchell. *Machine learning*. McGraw Hill, New York, 1996.

[MNS05] E. Macklovitch, N.T. Nguyen, and R. Silva. User evaluation report. Technical report, TransType2 (IST-2001-32091), 2005.

[Moo02] R.C. Moore. Fast and accurate sentence alignment of bilingual corpora. In *Proc. of AMTA'02*, pages 135–244, October 2002.

[Moo04] R.C. Moore. Improving IBM Word-Alignment Model 1. In *Proc. of ACL'04*, pages 519–524, July 2004.

[MW02] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *Proc. of EMNLP'02*, pages 133–139, Morristown, NJ, USA, July 2002. Association for Computational Linguistics.

[N$^+$92] S. Nirenburg et al. *Machine Translation: A Knowledge-based Approach*. Morgan Kaufmann, 1992.

[NCV03] F. Nevado, F. Casacuberta, and E. Vidal. Parallel corpora segmentation using anchor words. In *Proc. of EAMT/CLAW'03*, pages 33–40, May 2003.

[NGL97] H. T. Ng, W. B. Goh, and K. L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In Nicholas J. Belkin, A. Desai Narasimhalu, and Peter Willett, editors, *Proc. of SIGIR'97*, pages 67–73, Philadelphia, USA, 1997. ACM Press, New York, USA.

[NIS06] Nist 2006 machine translation evaluation official results. http://www.nist.gov/speech/tests/mt/mt06eval_official_results.html, November 2006.

[NM92] E. H. Nyberg and T. Mitamura. The kant system: fast, accurate, high-quality translation in practical domains. In *Proc. of CL'92*, pages 1069–1073, Morristown, NJ, USA, August 1992. Association for Computational Linguistics.

[Och03] F. J. Och. Minimum error rate training in statistical machine translation. In *Proc. of ACL'03*, pages 160–167, Morristown, NJ, USA, July 2003. Association for Computational Linguistics.

[ON03] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[ON04] F. J. Och and H. Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, 2004.

[P$^+$03] B. Pouliquen et al. Automatic annotation of multilingual text collections with a conceptual thesaurus. In *Proc. of EUROLAN'03*, July-August 2003.

[PJR07] D. Pinto, A. Juan, and P. Rosso. Using query-relevant documents pairs for cross-lingual information retrieval. In *Proc. of TSD'07*, pages 630–637, September 2007.

[PRWZ01] K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176, Thomas J. Watson Research Center, 2001.

[S$^+$06] M. Snover et al. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA'06*, pages 223–231, Boston, Massachusetts, USA, August 2006. Association for Machine Translation in the Americas.

[Seb02] F. Sebastiani. Machine learning in automated text categorisation. *ACM Computing Surveys*, 34(1):1–47, 2002.

[Seb06] Fabrizio Sebastiani. Classification of text, automatic. In Keith Brown, editor, *The Encyclopedia of Language and Linguistics*, volume 2, pages 457–463. Elsevier Science Publishers, Amsterdam, NL, second edition, 2006.

[SHP95] Hinrich Schütze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proc. of SIGIR'95*, pages 229–237, Seattle, US, July 1995. ACM Press, New York, US.

[Slo85] J. Slocum. A survey of machine translation: its history, current status and future prospects. *Computational Linguistics*, 11(1):1–17, 1985.

[Som99] H. L. Somers. Review article: Example-based machine translation. *Machine Translation*, 14(2):113–157, 1999.

[Som03] H. L. Sommers. *Computers and translation: a translator's guide*, chapter 3: Translation memory systems, pages 31–48. John Benjamins, 2003.

[SS00] R. E. Schapire and Y. Singer. Boostexter: A boosting-based systemfor text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

[SZH97] Z. Sen, Ch. Zhaoxiong, and H. Heyan. Interactive approach in machine translation systems. In *Proc. of IEEE ICPS'97*, pages 1814–1819, Beijing, China, October 1997.

[Tih82] B. Tihouin. The Meteo System. In Veronica Lawson, editor, *Proc. of Practical Experience of Machine Translation*, pages 39–44, 1982.

[TIM02] K. Toutanova, H. T. Ilhan, and C. D. Manning. Extensions to HMM-based statistical word alignment models. In *Proc. of EMNLP'02*, pages 87–94, Morristown, NJ, USA, July 2002. Association for Computational Linguistics.

[TJHA05] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[TN03] C. Tillmann and H. Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133, March 2003.

[Tom85] M. Tomita. Feasibility study of personal/interactive machine transaltion systems. In *Proc. of TMI'85*, pages 289–297, New York, USA, August 1985.

[TSM85] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.

[UM06] R. Udupa and H. K.r Maji. Computational complexity of statistical machine translation. In *Proc. of EACL'06*, April 2006.

[UN07] N. Ueffing and H. Ney. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40, 2007.

[V$^+$96] S. Vogel et al. HMM-based word alignment in statistical translation. In *Proc. of CL'96*, pages 836–841, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[VCR$^+$06] E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. Computer-assisted translation using speech recognition. *IEEE Transaction on Audio, Speech and Language Processing*, 14(3):941–951, 2006.

[Wea55] W. Weaver. Translation. In W. N. Locke and A. D. Booth, editors, *Machine Translation of Languages: fourteen essays*, pages 15–23. MIT Press, Cambridge, MA., 1955.

[WPW95] E. D. Wiener, J. O. Pedersen, and A. S. Weigend. A neural network approach to topic spotting. In *Proc. of SDAIR'95*, pages 317–332, April 1995.

[Wu83] C. F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[Wu96] D. Wu. A polynomial-time algorithm for statistical machine translation. In *Proc. of ACL'96*, pages 152–158, Morristown, NJ, USA, June 1996. Morgan Kaufmann / Association for Computational Linguistics.

[WW98] Y. Wang and A. Waibel. Fast decoding for statistical machine translation. In *Proc. of ICSLP'98*, pages 2775–2778, October 1998.

[WWC$^+$86] P. J. Whitelock, M. McGee Wood, B. J. Chandler, N. Holden, and H. J. Horsfall. Strategies for interactive machine translation: the experience and implications of the UMIST Japanese project. In *Proc. of COLING'86*, pages 329–334, Bonn, Germany, August 1986.

[Y$^+$93] J. Yamron et al. LINGSTAT: an interactive, machine-aided translation system. In *Proc. of HLT'93*, pages 191–195, Princeton, New Jersey, USA, March 1993.

[YK01] K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proc. of ACL'01*, pages 523–530, Morristown, NJ, USA, July 2001. Association for Computational Linguistics.

[Zaj88] R. Zajac. Interactive translation: a new approach. In *Proc. of COLING'88*, pages 785–790, Budapest, Hungary, August 1988.

CHAPTER 2

BILINGUAL TEXT CLASSIFICATION

2.1 Introduction

The proliferation of multilingual documentation in our Information Society has become a common phenomenon in many official institutions (EU parliament, the Canadian Parliament, UN sessions, Catalan and Basque Parliaments in Spain, etc.) and private companies (user's manuals, newspapers, books, etc.). In many cases, this textual information needs to be categorised by hand, entailing a time-consuming and arduous burden.

As mentioned in Section 1.2, monolingual TC has received most of the attention of the scientific community compared to bilingual (multilingual) or cross-lingual TC. Among the diverse approaches to monolingual TC the well-known naive Bayes classifier [Lew98, MN98] is one of the most popular. Being so, there have been several instantiations and generalisations of this classifier, from Bernoulli mixtures [JV02] to multinomial mixtures [N^{+}00, NM03]. Both generalisations seek to relax the naive Bayes feature independence assumption made when using a single Bernoulli or multinomial distribution per class.

The unrealistic assumption of the naive Bayes classifier is one of the main reasons explaining its comparatively poor results in contrast to other techniques such as *boosting-based classifier committees* (boosting) [SS00] and *support vector machines* (SVM) [BGV92, CV95, Vap95]. However, the performance of the naive Bayes classifier is significantly improved by using the generalisations mentioned above. Moreover, there are other recent generalisations (and corrections) that also overcome the weaknesses of the naive Bayes classifier and achieve very competitive results [SW02, R^{+}03, V^{+}04, P^{+}04a, P^{+}04b].

The accuracy of the single-class text classifiers presented above is considered to be fairly good [Seb02]. However, the performance of multi-label text classifiers is far from being acceptable, and it is more convenient to look at these classifiers as the backend of a MAI system [HH96, LH00, LD02, P^{+}03], as commented in

Section 1.2.2.

This chapter is devoted to bilingual TC, a novel application in the field of TC, that considers the case in which bilingual parallel texts are to be classified. The organisation of this chapter is as follows. We first introduce the unigram mixture model[a] and its maximum likelihood estimation in Section 2.2, performing analogously with the bilingual unigram mixture model in Section 2.3. Then, we derive five bilingual text classifiers grounded on the unigram and bilingual unigram models in Section 2.4. The presentation of these bilingual classifiers is followed in Section 2.5 by a series of experimental results on three tasks of different complexity, together with comparative results SVM and boosting techniques. Finally, we state the conclusions and future work in Section 2.6.

2.2 Unigram mixture model

2.2.1 The model

Let us consider the p.f. over sequences of words of the form $x = x_1 \dots x_j \dots x_{|x|}$ of known length $|x|$

$$p(x) = \prod_{j=1}^{|x|} p(x_j \mid x_1^{j-1}). \tag{2.1}$$

For the unigram model, we assume that the probability of each word to occur does not depend on any previous word[b],

$$p(x_j \mid x_1^{j-1}) := p(x_j). \tag{2.2}$$

Thus, the unigram model becomes

$$p(x; \mathbf{\Theta}) = \prod_{j=1}^{|x|} p(x_j). \tag{2.3}$$

where

$$\mathbf{\Theta} = (p(u) : u \in \mathcal{X}) \,. \tag{2.4}$$

A *unigram mixture model* is an instance of the general mixture model defined in Eq. (1.41), where $p(x \mid t; \mathbf{\Theta}_t)$ is a component-dependent version of the unigram model presented in Eq. (2.3)

$$p(x \mid t; \mathbf{\Theta}_t) = \prod_{j=1}^{|x|} p(x_j \mid t). \tag{2.5}$$

[a] A unigram language model is just a multinomial word distribution.

[b] We do not distinguish between the general probability function and the model itself, since it is clear by the context or the introduction of the parameter vector $\mathbf{\Theta}$.

where each component has its own vector of unigrams

$$\mathbf{\Theta}_t = (p(u \,|\, t) : u \in \mathcal{X}) \tag{2.6}$$

being $\mathcal{X}$, the vocabulary from which the word u is drawn and $p(u \,|\, t)$, the probability of word u to occur at component t.

2.2.2 Maximum likelihood estimation

Let $X = (x_1, \ldots, x_N)^t$ be a set of samples available to learn the unigram mixture model presented in Section 2.2.1. Following the maximum likelihood principle, optimal parameter values maximise the log-likelihood function of $\mathbf{\Theta}$

$$L(\mathbf{\Theta}; X) = \sum_{n=1}^{N} \log \sum_{\boldsymbol{z}_n} \prod_{t=1}^{T} [p(t)\, p(x_n \,|\, t; \mathbf{\Theta}_t)]^{z_{nt}} \tag{2.7}$$

as in Eq. (1.42). Here, we consider an specific instantiation of the EM algorithm for mixture models presented in Section 1.9, for the unigram mixture model. The Q function is defined as in Eq. (1.44), with $z_{nt}^{(k)}$ in Eq. (1.45) being the posterior probability of x_n being actually generated by the tth component-conditional unigram model, as defined in Eq. (2.5).

In the M step we compute Eq. (1.28), in order to find a new estimate for the mixture coefficients as in Eq. (1.48), and for the component-conditional unigram parameters,

$$p(u \,|\, t)^{(k+1)} = \frac{N(u,t)}{\sum\limits_{u' \in \mathcal{X}} N(u',t)} \quad \forall t, u \in \mathcal{X} \tag{2.8}$$

where

$$N(u,t) = \sum_{n=1}^{N} z_{nt}^{(k)} \sum_{j=1}^{|x_n|} \delta(x_{nj} = u) \tag{2.9}$$

and

$$\delta(b) = \begin{cases} 1 & \text{if } b \text{ is true} \\ 0 & \text{if } b \text{ is false.} \end{cases} \tag{2.10}$$

The δ function will be used throughout this thesis to simplify the mathematical notation. Eq. (2.8) can be understood as weighted relative counts of unigrams for each component t, in which the weighting term z_{nt} accounts for how much the nth sample contributes to the counts of the tth component. The asymptotic cost of the training algorithm per iteration is $O(N \cdot T \cdot \overline{|x|})$, where $\overline{|x|}$ is the sentence average length.

2.3 Bilingual unigram mixture model

2.3.1 The model

Bilingual texts are pairs of sentences or documents (x, y) that are mutual translations, i.e. x is a sentence in a source language, and y is its corresponding translation in a target language. In this section, we present a bilingual mixture model based on the unigram distribution to tackle the modelisation of bilingual texts.

To this purpose, let us consider the joint p.f. over x and y,

$$p(x, y; \mathbf{\Theta}) = \sum_{\boldsymbol{z}} \prod_{t=1}^{T} [p(t)\, p(x, y \mid t; \mathbf{\Theta}_t)]^{z_t} \tag{2.11}$$

where, similarly to Eq. (1.41), $p(t)$ and $p(x, y \mid t; \mathbf{\Theta}_t)$ are the mixture coefficient and the component-conditional p.f. of the tth component, respectively. In what follows, we assume x and y to be two conditionally independent variables given the mixture component t from which they were drawn

$$p(x, y \mid t; \mathbf{\Theta}_t) = p(x \mid t; \mathbf{\Theta}_t)\, p(y \mid t; \mathbf{\Theta}_t). \tag{2.12}$$

Plugging Eq. (2.12) into Eq. (2.11) results in the *bilingual unigram mixture model*

$$p(x, y; \mathbf{\Theta}) = \sum_{\boldsymbol{z}} \prod_{t=1}^{T} [p(t)\, p(x \mid t; \mathbf{\Theta}_t)\, p(y \mid t; \mathbf{\Theta}_t)]^{z_t} \tag{2.13}$$

where $p(x \mid t; \mathbf{\Theta}_t)$ is the component-dependent unigram model for the source language as defined in Eq. (2.5), and $p(y \mid t; \mathbf{\Theta}_t)$ is the component-dependent unigram model for the target language

$$p(y \mid t; \mathbf{\Theta}_t) = \prod_{i=1}^{|y|} p(y_i \mid t). \tag{2.14}$$

Thus, the global vector of parameters $\mathbf{\Theta}$ is of the form of Eq. (1.34) where

$$\mathbf{\Theta}_t = \left\{ \begin{array}{ll} p(u \mid t) & u \in \mathcal{X} \\ p(v \mid t) & v \in \mathcal{Y} \end{array} \right. . \tag{2.15}$$

is the source and target language unigrams, respectively.

2.3.2 Maximum likelihood estimation

The maximum likelihood estimation of this model is similar to that of the unigram mixture model in Section 2.2.2. Generally speaking, this reduces to substituting x_n by (x_n, y_n).

Let $(X, Y) = ((x_1, y_1), \ldots, (x_N, y_N))^t$ be a set of samples available to learn $\mathbf{\Theta}$. The optimal parameter values maximise the log-likelihood function of $\mathbf{\Theta}$ w.r.t. (X, Y), that can be expressed in terms of the indicator vector $\boldsymbol{z}_n$ as

$$L(\mathbf{\Theta}; X, Y) = \sum_{n=1}^{N} \log \sum_{\boldsymbol{z}_n} \prod_{t=1}^{T} [p(t)\, p(x_n \mid t; \mathbf{\Theta}_t)\, p(y_n \mid t; \mathbf{\Theta}_t)]^{z_{nt}} \tag{2.16}$$

As shown in Section 1.9 for finite mixtures in general, we revert to the EM algorithm to obtain a maximum likelihood estimation of $\mathbf{\Theta}$. To this purpose, we replace x_n by (x_n, y_n) in Eq. (1.44) to compute the expected value of $p(X, Y, Z; \mathbf{\Theta})$ w.r.t. the posterior $p(Z \mid X, Y; \mathbf{\Theta}^{(k)})$,

$$Q(\mathbf{\Theta} \mid \mathbf{\Theta}^{(k)}) = \sum_{n=1}^{N} \sum_{t=1}^{T} z_{nt}^{(k)} \left[\log p(t) + \log p(x \mid t; \mathbf{\Theta}_t) + \log p(y \mid t; \mathbf{\Theta}_t)\right] \tag{2.17}$$

where

$$z_{nt}^{(k)} = \frac{p(t)^{(k)}\, p(x \mid t; \mathbf{\Theta}_t^{(k)})\, p(y \mid t; \mathbf{\Theta}_t^{(k)})}{\sum_{t'=1}^{T} p(t')^{(k)}\, p(x \mid t; \mathbf{\Theta}_t^{(k)})\, p(y \mid t; \mathbf{\Theta}_t^{(k)})}\,. \tag{2.18}$$

In the M step, equivalently to Eq. (1.46), we maximise Eq. (2.17) so as to find a new estimate for $\mathbf{\Theta}$, $\mathbf{\Theta}^{(k+1)}$. This results in updating equations that are analogous to those in Section 2.2.2. Specifically, the mixture coefficients are calculated as in Eq. (1.48) and the source-language component-conditional unigram parameters as in Eq. (2.8). The update equation for the target-language component-conditional unigram parameters is

$$p(v \mid t)^{(k+1)} = \frac{N(v, t)}{\sum\limits_{v' \in \mathcal{X}} N(v', t)} \quad \forall t, v \in \mathcal{Y} \tag{2.19}$$

where

$$N(v, t) = \sum_{n=1}^{N} z_{nt}^{(k)} \sum_{i=1}^{|y_n|} \delta(y_{ni} = v)\,. \tag{2.20}$$

The asymptotic cost of the training algorithm per iteration is $O(N \cdot T \cdot (\overline{|x|} + \overline{|y|}))$, where $\overline{|x|}$ and $\overline{|y|}$ are the source and target average lengths, respectively.

2.3.3 Smoothing

Two major problems in parameter estimation are zero probabilities due to model overfitting, and the occurrence of infrequent events whose probabilities are poorly estimated. The usual solution to this problem is what is known in the literature as *smoothing*. Smoothing basically consists in the interpolation of a specific probability distribution, which we are estimating in our training process, with a more general distribution.

In this chapter, we smooth the component-conditional unigram distribution after each M step, interpolating the estimated probability distribution $p(u \mid t)$ with a uniform probability distribution

$$\hat{p}(u \mid t) = (1 - \epsilon)\, p(u \mid t) + \epsilon\, \frac{1}{|\mathcal{X}|} \tag{2.21}$$

where ϵ is an interpolation parameter that weighs the contribution of each distribution. A different interpretation of the ϵ parameter would be the amount of probability mass that we discount from the specific distribution to be uniformly shared among all the words. The ϵ parameter was manually fixed in order to obtain smoothed CER and log-likelihood curves as we increase the number of components in the mixture model.

2.4 Bilingual text classification using unigram models

The mixture models explained in Sections 2.2 and 2.3 are the basis for supervised bilingual text classifiers depicted in this section.

2.4.1 Decision rules

Let us consider the task in which we have to classify a bilingual pair (x, y) into one of C supervised classes. As shown in Section 1.2, the optimal classification decision is the Bayes rule that assigns (x, y) to a class with maximum posterior probability. The Bayes rule in Eq. 1.4 requires a class-conditional p.f. that in our case is instantiated in $p(x, y \mid c)$. This fact involves the definition of supervised class-conditional versions of the unigram mixture model in Section 2.2 and the bilingual unigram mixture model in Section 2.3.

In this thesis we consider five bilingual classification rules, the first four of them are based on the unigram mixture model and the last one, on the bilingual unigram mixture model. Firstly, the monolingual source-language rule simply ignores the target text

$$c(x, y) = \arg\max_{c} \; \log p(c) + \log \sum_{t=1}^{T} p(t \mid c)\, p(x \mid t, c; \mathbf{\Theta}_{ct}). \tag{2.22}$$

A similar rule holds for the monolingual target-language model. Alternatively we could think of a unigram model trained on the concatenation of source and target texts

$$c(xy) = \arg\max_{c} \; \log p(c) + \log \sum_{t=1}^{T} p(t \mid c)\, p(xy \mid t, c; \mathbf{\Theta}_{ct}) \tag{2.23}$$

where xy represents the concatenation of the source and target texts. This model is referred as to the *bilingual bag-of-words* (BBoW) model in this thesis.

Furthermore, we could carry out a *global* decomposition of the bilingual p.f. into two unigram mixture p.f.

$$p(x,y) = \sum_{t=1}^{T} p(t)\, p(x \mid t) \sum_{t'=1}^{T} p(t')\, p(y \mid t') \tag{2.24}$$

so the classification rule becomes,

$$\begin{aligned} c(x,y) = \arg\max_{c} \log p(c) + \, \log \sum_{t=1}^{T} p(t \mid c)\, p(x \mid t, c; \mathbf{\Theta}_{ct}) \, + \\ + \, \log \sum_{t'=1}^{T} p(t' \mid c)\, p(y \mid t', c; \mathbf{\Theta}_{ct}). \end{aligned} \tag{2.25}$$

Finally, a *local* decomposition of the bilingual p.f. yields a classification rule based on the bilingual unigram mixture model

$$c(x,y) = \arg\max_{c} \log p(c) + \sum_{t=1}^{T} p(t \mid c)\, p(x \mid t, c; \mathbf{\Theta}_{ct})\, p(y \mid t, c; \mathbf{\Theta}_{ct}). \tag{2.26}$$

2.4.2 Maximum likelihood estimation for supervised classification

The unigram-based models can be used as class-conditional models in supervised classification. To this purpose, we can extend the E and M steps of the EM algorithm in order to carry out the training process for several supervised classes simultaneously. This simple extension of the EM algorithm is equivalent to the usual practise of applying its basic version to each supervised class in turn. However, we prefer to adopt the extended EM, mainly to have a unified framework for classifier training in accordance with the log-likelihood criterion.

For the sake of simplicity, we just present the derivation for the unigram mixture model. In a supervised setting, training samples come with their corresponding class labels, $(X,C) = ((x_1,c_1),\ldots,(x_N,c_N))^t$, and the vector of unknown parameters is

$$\mathbf{\Psi} = (p(1),\ldots,p(C); \mathbf{\Theta}_1,\ldots,\mathbf{\Theta}_C) \tag{2.27}$$

where, for each supervised class c, its prior probability is given by $p(c)$ and its class-conditional probability function is a unigram mixture controlled by a vector of the form of Eq. (1.34), $\mathbf{\Theta}_c$. The log-likelihood of $\mathbf{\Psi}$ w.r.t. the labelled data is

$$L(\mathbf{\Psi}; X, C) = \sum_{c=1}^{C} N_c \, \log p(c) + L_c(\mathbf{\Theta}_c; X_c) \tag{2.28}$$

where X_c is the set of samples in X labelled as belonging to class c, and N_c is the number of samples in X_c. The function L_c is as the log-likelihood function in

Eq. (2.7), but only for the parameter vector of class c, $\boldsymbol{\Theta}_c$, w.r.t. the samples in class c,

$$L_c(\boldsymbol{\Theta}_c; X_c) = \sum_{n=1}^{N} \delta(c_n = c) \, \log \sum_{\boldsymbol{z}_n} \prod_{t=1}^{T} [p(t \mid c_n) \, p(x_n \mid t, c_n; \boldsymbol{\Theta}_{c_n t})]^{z_{nt}} \tag{2.29}$$

which can be optimised by a simple extension of the EM algorithm given in Section 2.2.2.

More precisely, the E step computes Eq. (1.45) using $\boldsymbol{\Theta}_{c_n}$ for those x_n belonging to class c_n,

$$z_{nt}^{(k)} = \frac{p(t \mid c_n)^{(k)} \, p(x_n \mid t, c_n; \boldsymbol{\Theta}_{c_n t})^{(k)}}{\sum_{t'=1}^{T} p(t' \mid c_n)^{(k)} \; p(x_n \mid t', c_n; \boldsymbol{\Theta}_{c_n t'})^{(k)}} . \tag{2.30}$$

The M step computes the conventional estimates for class priors,

$$p(c) = \frac{N_c}{N} \qquad \forall c, \tag{2.31}$$

class-dependent versions of the update equation for mixture coefficients in Eq. (1.48),

$$p(t \mid c)^{(k+1)} = \frac{1}{N} \sum_{n=1}^{N} z_{nt}^{(k)} \, \delta(c_n = c) \qquad \forall c, t \tag{2.32}$$

and class-dependent versions of the update equation for component-conditional unigrams in Eq.(2.8),

$$p(u \mid t, c)^{(k+1)} = \frac{N(u, t, c)}{\sum_{u' \in \mathcal{X}} N(u', t, c)} \quad \forall c, t, u \in \mathcal{X} \tag{2.33}$$

where

$$N(u, t, c) = \sum_{n=1}^{N} \delta(c_n = c) \, z_{nt}^{(k)} \sum_{j=1}^{|x_n|} \delta(x_{nj} = u) \, . \tag{2.34}$$

Note that the estimation computed in Eq. (2.31) is invariant over the estimation process.

2.5 Experimental results

The five models considered were assessed and compared on three bilingual text classification tasks known as the Traveller dataset, the BAF (French acronym for English-French Bitext) corpus and the JRC-Acquis corpus. This section first describes these datasets and then provides the experimental results obtained on them.

2.5.1 Datasets

Traveller dataset

The Traveller dataset comes from a limited-domain Spanish-English machine translation application for human-to-human communication situations in the front-desk of a hotel [ABC$^+$00]. It was semi-automatically built from a small "seed" dataset of sentence pairs collected from traveller-oriented booklets by four persons; A, F, J and P, in accordance with the subdomain assignment given in Table 2.1. Note that each person had to cater for a (non-disjoint) subset of subdomains, and thus each person can be considered a different (multimodal) class of Spanish-English sentence pairs. Therefore, the task will be to classify sentence pairs into one of the four disjoint classes A, F, J or P, that is, to identify the authorship of a given sentence pair. Subdomain overlapping among classes would foresee that perfect classification is not possible, although in our case, low CER will indicate that our mixture model has been able to capture the multimodal nature of the data in each class. Unfortunately, the subdomain of each pair was not recorded, and hence we cannot train a subdomain-supervised unigram mixture in each class to see how it compares to mixtures learnt without such supervision.

The Traveller dataset contains $8,000$ sentence pairs, with $2,000$ pairs per class. The size of the vocabulary and the number of singletons reflect the relative simplicity of this corpus. Some statistics are shown in Table 2.2.

Table 2.1: Subdomain assignment in the Traveller dataset.

Persons				Subdomain	
A	F	J	P	#	Description
✓	✓			1	notifying a previous reservation
✓				2	asking about rooms
✓				3	having a look at rooms
✓	✓			4	asking for rooms
✓				5	signing the registration form
✓				6	complaining about rooms
✓				7	changing rooms
	✓			8	asking for wake-up calls
	✓			9	asking for keys
	✓	✓		10	asking for moving the luggage
		✓		11	notifying the departure
			✓	12	asking for the bill
		✓	✓	13	asking about the bill
			✓	14	complaining about the bill
			✓	15	asking for a taxi
✓	✓	✓	✓	16	general sentences

BAF corpus

The BAF corpus[c] [Sim98] is a compilation of bilingual "institutional" French-English texts ranging from debates of the Canadian parliament (Hansard), court transcripts and UN reports to scientific, technical and literary documents. This dataset contains 11 documents trying to be representative of the types of text that are available in multilingual versions. They are organised into 4 natural disjoint genres: *institutional*, *scientific*, *technical* and *literary*. *Institutional* and *scientific* classes comprises documents from the original pool of 11 documents, which were theme-related, but devoted to heterogeneous purposes or written by different authors. This fact provides the multimodal nature to the BAF corpus that can be adequately modelled by mixture models. The BAF corpus was aligned at the sentence level by human experts. It was initially thought to be used as a reference corpus to evaluate automatic alignment techniques in machine translation.

Prior to performing the experiments, the BAF corpus was simplified in order to reduce the size of the vocabulary and discard spurious sentence pairs. This preprocessing mainly consisted in three basic actions: downcasing, replacement of those words containing a sequence of numbers by a generic label, and isolation of punctuation marks. This basic procedure halved the size of the vocabulary and significantly simplified this corpus. Neither stopword lists, nor stemming techniques were applied since, as shown in [V$^+$04], it is unclear whether this further preprocessing may be convenient. As seen in Table 2.2, this corpus is much more complex than the Traveller dataset.

JRC-Acquis corpus

The JRC-Acquis corpus[d][SPW$^+$06] is a multilingual parallel corpus in more than 20 languages containing documents extracted from the Acquis Communautaire that constitutes the body of common rights and obligations which bind all the Member States together within the European Union.

Like most other official documents of the European Commission and the European Parliament, the Acquis texts have been classified according to the multilingual, hierarchically organised EuroVoc thesaurus [EC95], which is a classification system with over 6,000 hierarchically organised classes. The main subject domains assigned to the document collection cover economy, health, information technology, law, agriculture, food, politics and more. However, each document receives a variable number of specific descriptors (class labels) that are found at the lowest level of the Eurovoc hierarchy, reflecting the multi-topic nature of these documents.

The JRC-Acquis corpus was aligned at the paragraph level using the Vanilla aligner[e] which implements the Gale and Church alignment algorithm [GC93], and

[c] Available at http://rali.iro.umontreal.ca/Ressources/BAF

[d] Version 2.2. employed in this thesis is available at http://wt.jrc.it/lt/Acquis

[e] Available at http://nl.ijs.si/telri/Vanilla

Table 2.2: Traveller task (top) and the BAF (bottom) corpus statistics, for all classes, respectively, in the form of ”Spanish/English” for Traveller and ”French/English” for BAF. Abbreviations used: SENTS=sentences; AVGSLEN=average sentence length; RUNKW=running Kilo-words; VOCAB=vocabulary size; and STONS =singletons;

Traveller task

CLASS	A	F	J	P	ALL
SENTS	2,000	2,000	2,000	2,000	8,000
RUNKW	19/17	25/23	20/17	21/22	86/80
VOCAB	329/311	442/317	179/140	142/51	679/503
AVGSLEN	9/8	12/11	10/8	10/11	10/10
STONS	95/106	78/71	0/0	1/0	47/43

BAF corpus

CLASS	INST	LITE	SCIE	TECH	ALL
SENTS	10,988	2,435	2,295	3,021	18,739
RUNKW	351/301	56/51	61/53	48/40	516/445
VOCAB	14,046/10,858	7,124/5,607	5,975/4,997	2,542/2,053	20,454/15,471
AVGSLEN	31/27	22/20	26/23	16/13	27/23
STONS	5,329/3,709	4,034/2,548	2,791/2,099	939/628	8,205/5,353

the HunAlign aligner [V$^+$05]. However we only made use of the alignment information at the document level, since our task is descriptor assignment at that level.

Before training our text classifiers, the JRC-Acquis corpus underwent the same basic preprocessing as the BAF corpus. Here we also preferred not to apply any language-dependent preprocessing such as stemming techniques or stopword lists.

In our experiments, we only used those documents drawn from the French-English partition of the JRC-Acquis corpus, retaining those descriptors occurring at least 5 times. As a result, there are 990 different descriptors in this partition. Some statistics of the preprocessed French-English partition of this corpus are shown in Table 2.3. Comparing these figures to those of the BAF corpus, we can observe that the ratio between singletons and vocabulary is similar, while the number of running words and the average length is about two orders of magnitude superior in the case of the JRC-Acquis corpus. This longer average length comes to compensate somehow the scarcity of samples per class. However, the number of classes in the JRC-Acquis is almost three orders of magnitude greater than that of the BAF corpus.

2.5.2 Experimental results on Traveller and BAF

A series of experiments were carried out to analyse the behaviour of each individual classifier in terms of log-likelihood and CER as a function of the number of mixture components per class ($T \in \{1, 2, 5, 10, 20, 50, 100\}$). This was done for a training

	French	English
Number of documents	5108	
Average length (in words)	1,819	1,564
Vocabulary	36.6K	32.5K
Singletons	10.6K	10.5K
Running words	9.3M	8.0M

Table 2.3: Basic statistics of the preprocessed French-English partition of the JRC-Acquis corpus ($K = \times 10^3$ and $M = \times 10^6$).

and test sets resulting from a random dataset partition (1/2-1/2 split for Traveller and 4/5-1/5 for BAF).

Figure 2.1 shows the evolution of CER (left y axis) and log-likelihood (right y axis), on training and test sets, for an increasing number of mixture components (x axis). From top to bottom rows we have: the best monolingual classifier (English in both datasets), the BBoW classifier, and global and local classifiers. Each plotted point is an average over values obtained from 30 randomised trials.

From the results in Figure 2.1, we can see that the evolution of the log-likelihood on the training and test sets is as theoretically expected, for all classifiers in both, Traveller and BAF. The log-likelihood in training always increases, while the log-likelihood in test increases up to a moderate number of components ($20 - 50$ in Traveller and $5 - 10$ in BAF). This number of components can be considered as an indication of the number of "natural" subclasses in the data. About this number of mixture components is also commonly found the lowest classification test error rate, as it occurs in our case. As the number of components keeps increasing, the well-known overtraining effect appears, the log-likelihood in test falls and the accuracy degrades. For this reason we decided to limit the number of mixture components to 100. Additional informal trials (not reported here) with an increasing number of mixture components confirmed this performance degradation.

Figure 2.2 shows competing curves for test error-rate as a function of the number of mixture components for the English-based, BBoW, global and local classifiers; there are two plots, one for Traveller and the other for BAF. Error bars representing 95% confidence intervals are plotted for the English-based classifiers in both plots, and the global classifier in BAF.

From the results for Traveller in Figure 2.2, we can see that there is no significant statistical difference in terms of error rate between the best monolingual classifier and the bilingual classifiers. The reason behind these similar results can be better explained in the light of the statistics of the Traveller dataset shown in Table 2.2. The simplicity of the Traveller dataset, characterised by its small vocabulary size and its large number of running words, allows for a reliable estimation of model parameters in both languages. This is reflected in the high accuracy (~ 95%) of the monolingual classifiers and the little contribution of a second language to

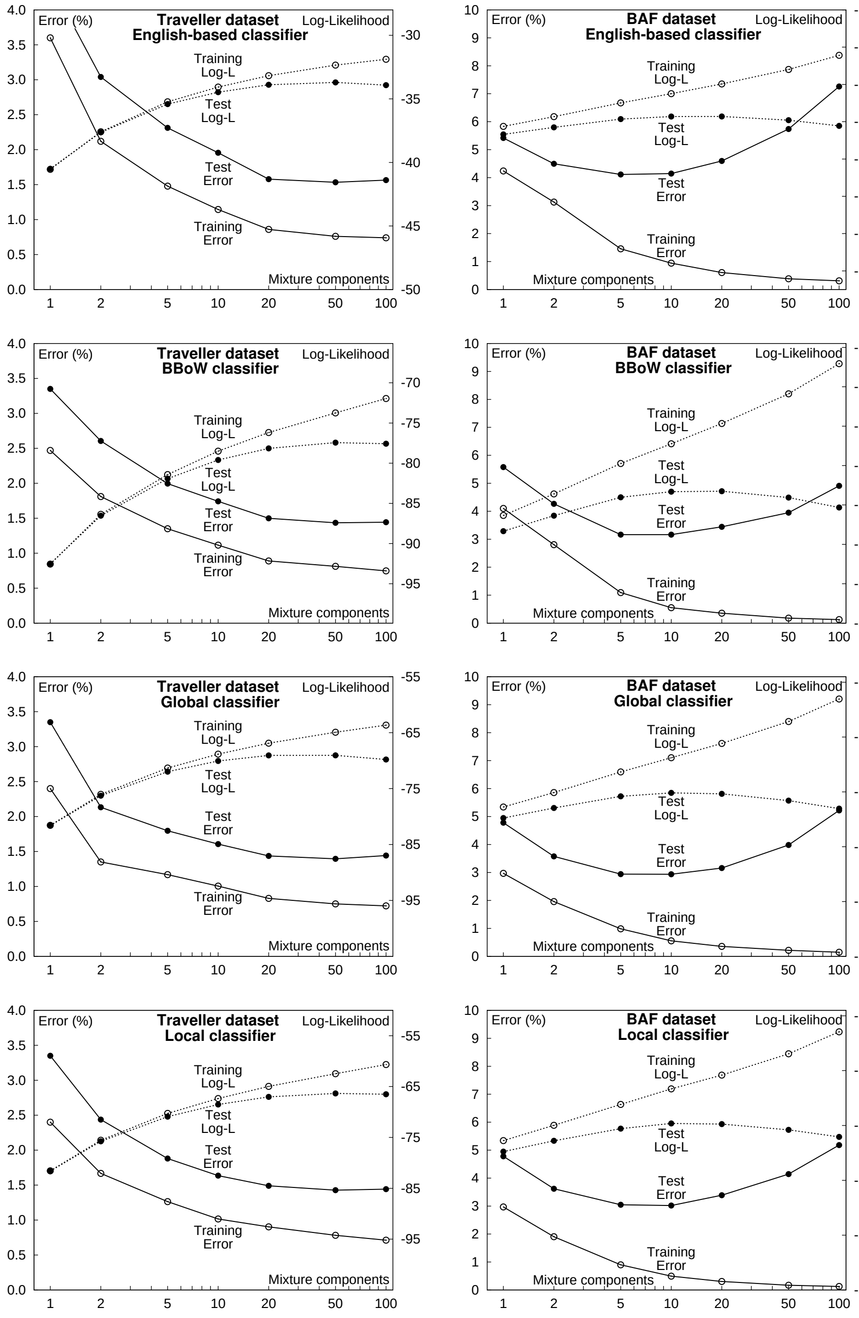


Figure 2.1: Error rate and log-likelihood curves in training and test sets as a function of the number of mixture components, in Traveller (left column) and BAF (right column) for the four classifiers considered. Classifiers: the best monolingual, the BBoW, the global and the local classifier.

boost the performance of bilingual classifiers. Nevertheless, bilingual classifiers seem to achieve systematically better results.

In contrast to the results obtained for Traveller, the results for BAF in Figure 2.2 indicate that bilingual classifiers perform significantly better than monolingual models. More precisely, if we compare the curves for the English-based classifier and the global classifier, we can observe that there is no overlapping between their error-rate confidence intervals. Clearly, the complexity and data scarcity problem of the BAF corpus lead to poorly estimated models, favouring bilingual classifiers that take advantage of both languages. However, the different bilingual classifiers have similar performance.

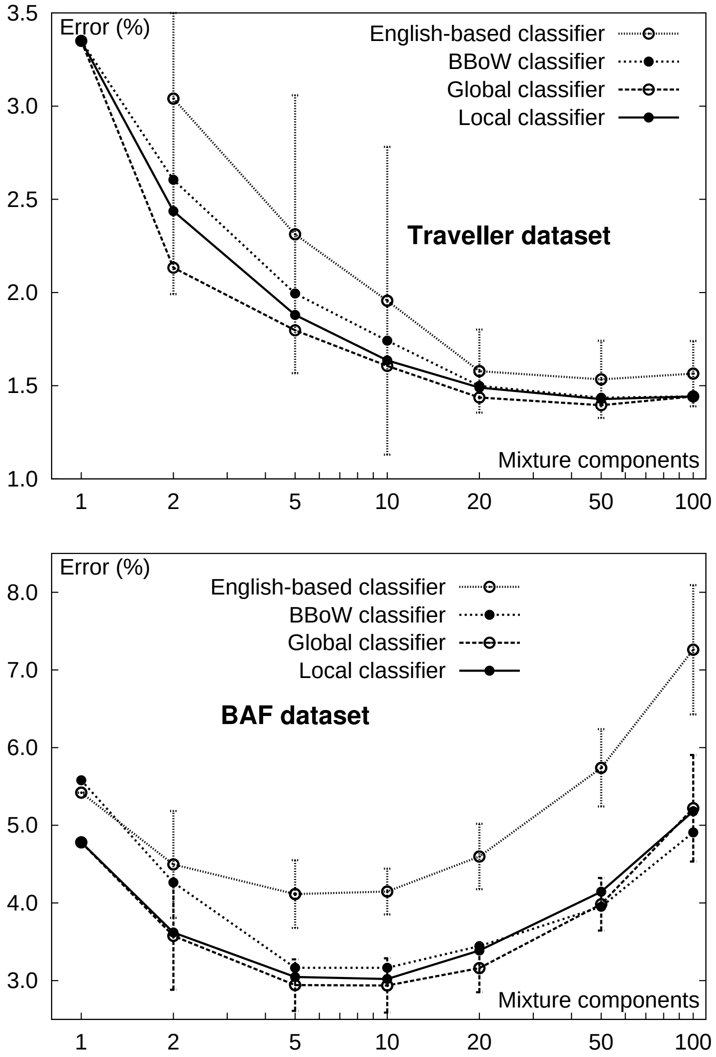


Figure 2.2: Test-set error rate curves as a function of the number of mixture components, for each classifier in Traveller and BAF

2.5.3 Experimental results on JRC-Acquis

In the previous section, we evaluated our bilingual models on two single-class classification tasks, the Traveller task and the BAF corpus. In the case of the JRC-Acquis corpus, we tackle a multi-label classification problem from the viewpoint of a MAI tool as presented in Section 1.2.2.

The evaluation was performed using the bilingual local and the English monolingual classifiers, on random 80%-20% train-test splits of the French-English JRC-Acquis partition. In this task, the average computing time[f] for the bilingual local classifier is 8 minutes per iteration and component.

Also, it should be noticed that the number of EuroVoc descriptors varies from one document to other, so a strategy to select the right number of descriptors for each document is required. In Figure 2.3, we have simply extracted five descriptors per document, which is the average number of descriptors in the whole corpus. This figure shows macro-averaging precision and recall curves as a function of the number of mixture components per class, for the best monolingual (English-only) and the bilingual local classifier. Each plotted point is an average over values obtained from 6 randomised trials.

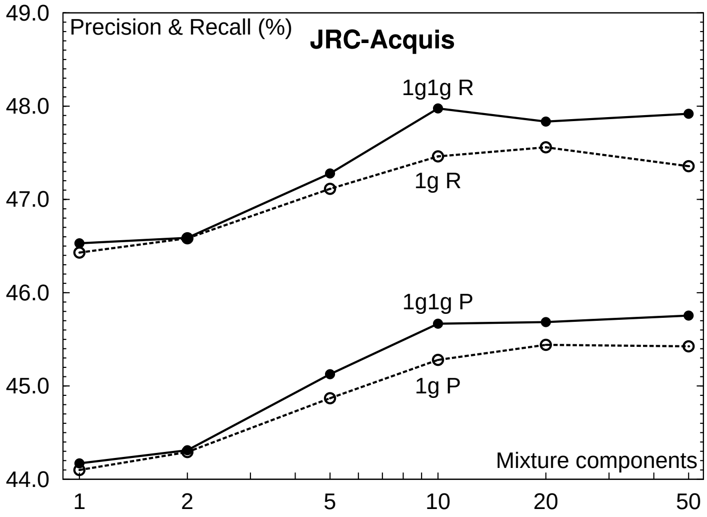


Figure 2.3: Macro-averaging precision (P) and recall (R) curves as a function of the number of mixture components (x axis) for the English-only ($1g$) and bilingual ($1g1g$) unigram mixture classifiers.

From the results in Figure 2.3 clearly outstand the benefits of multiple component over single component modelling. This fact is statistically significant when we use 5 or more components. However, the figures of bilingual classifiers are not statistically significantly better than those of their monolingual counterparts. We

[f] On a 2.0 GHz Intel Xeon machine

Number of labels	1	5	10	$\|R\|$
Precision	62.0	45.7	31.1	46.7
Recall	13.4	47.8	64.3	46.7

Table 2.4: Macro-averaging precision and recall figures as a function of the number of labels offered by the MAI system.

believe this is because the length of the texts in the JRC-Acquis corpus is two orders of magnitude longer than in *Traveller* and *BAF* corpora. Therefore, it seems that the additional information of the second language is not so essential as it is in other tasks with shorter texts, in which the contribution of a second language is more significant.

Table 2.4 presents macro-averaging precision and recall as a function of the number of labels offered by the MAI system. $|R|$ stands for the number of labels of the test sample, i.e., an ideal fictitious scenario in which the MAI system always provides the right number of labels for each document. For this reason, precision and recall are equal. As expected, precision degrades while recall improves, as we increase the number of labels.

The excellence of these results should be assessed bearing in mind the complexity of this task and how MAI systems work. On the one hand, professional indexers do not completely agree on the most suitable descriptors for a given document. Indeed, previous studies [P$^+$03] on annotator agreement maintain that keyword overlapping among indexers is about 70% to 80%.

On the other hand, MAI systems work by providing a lengthy list of descriptors from which an indexer would select those ones considered most appropriated. For evaluation purposes we decided that our MAI system should provide only 5 descriptors for each document, seeking a balance between precision and recall. However, in a MAI scenario, we would be more interested in recall since we would like that our system provides a longer list of descriptors, from which a indexer would filter out those unsuitable descriptors.

Taking this into account, the figures in Figure 2.4 revealed that our MAI system would be offering up to 64.3% of the correct descriptors for a list of 10 descriptors. This figure conveys the possibility of a MAI system which suggests many of the desired descriptors.

2.5.4 Comparative results

In this section, we compare the performance of the bilingual local classifier to that of state-of-the-art techniques in TC. More precisely, we study SVM implemented in the SVM^{light} toolkit and boosting instantiated in BoosTexter.

Regarding the experiments with SVM, we focused on the conventional vector-based kernels, however some authors have proposed the use of string-based kernels. In string kernels, the conventional term frequency features in vector-based

Table 2.5: Test-set CERs (in percentage) on Traveller and BAF, and macro-averaging precision and recall figures on JRC-Acquis for the bilingual local decomposition, SVM^{light} and BoosTexter classifiers.

	Traveller	BAF	JRC-Acquis	
	CER		precision	recall
Bilingual Local	1.4	3.0	45.9	47.7
SVM^{light}	1.5	9.0	N/A	N/A
BoosTexter	1.2	5.8	43.2	44.9

kernels are replaced by all possible ordered subsequences of characters or words in the document. While the well-known dot product is substituted by a string similarity measure efficiently computed with a dynamic programming algorithm [Wat00, LSST^{+}02, CGGR03].

The experiments reported with SVM were carried out with linear kernel functions, although the polynomial kernels were informally evaluated with poorer results. As regards the feature representation, we utilised unigram (term) frequency. For BoosTexter, we trained decision trees (weak learner) on the presence or absence of unigrams. SVM and boosting classifiers were trained on the bilingual training set resulting of the concatenation of source and target sentences, as we did in the BBoW model .

In order to tune the different parameters of SVM^{light} and BoosTexter, we performed a 10-fold cross-validation on the training set of the Traveller task and BAF corpus, although this tuning procedure was not feasible for the JRC-Acquis due to excessive running time. In the latter corpus, for BoosTexter, we also considered as objective function *ranking* instead of Hamming loss, but the results were inferior. Once these parameters were adjusted, we run the same configuration on the test set.

The results on the test set are shown in Table 2.5. SVM^{light} and BoosTexter offer similar performance to the bilingual local classifier on the Traveller, although BoosTexter achieves the best result on this task. However, the bilingual local classifier statistically significantly outperforms SVM^{light} and BoosTexter on the BAF corpus. On the JRC-Acquis task, the bilingual local classifier obtained better precision and recall figures than BoosTexter. However, it was not possible to run experiments with SVM^{light} on this corpus due to memory constraints.

2.6 Conclusions and future work

We have presented three extensions of the unigram mixture-based model for bilingual text: the BBoW model, and the global and local decomposition models. The performance of these extensions was compared to that of SVM and boosting classifiers.

Two outstanding conclusions can be stated from the results presented in this chapter. First, mixture-based classifiers surpass single-component classifiers in all cases (monolingual, BBoW, global and local). In fact, we have taken advantage of the flexibility of the mixture modelisation over the single-component approach to further improve the error rates achieved. Second, bilingual classifiers outperform their monolingual counterparts in the Traveller task and the BAF corpus. However, this is not the case on the JRC-Acquis corpus in which the monolingual and bilingual classifiers show similar performance. This leads us to think that the incorporation of a second language into a text classifier is significant in the presence of data scarcity. In the Traveller task and BAF corpus we have only 10-12 and 20-30 words per sentence on average, respectively. So, if we duplicate this small number of words by adding a second language, it notably helps to improve the accuracy of the classifier. However, in the JRC-Acquis the average length is 1,500-1,800 words per document, so it seems that the contribution of the second language is not so significant.

Moreover, we have compared the performance of our bilingual unigram-based classifier to state-of-the-art techniques in TC: SVM and boosting. We have observed that the bilingual local classifier obtains similar results to these latter techniques on the Traveller task, but it outperforms them on the BAF corpus. This is also the case on the JRC-Acquis.

Furthermore, the accuracy of the bilingual local classifier on the JRC-Acquis is good enough to support a MAI system, that would be providing on average about 65% of the correct descriptors associated with a document.

A direct extension of the models presented in this chapter would be to go beyond the unigram representation and take advantage of the context information [SW02, P$^+$04b]. In appendix A, we scratch the surface of this approach by considering smoothed n-gram language models and n-gram features to train SVM and boosting methods.

As future work, we plan to investigate the application of mixture modelling to smoothed n-gram models which has been successfully tested in automatic speech recognition [IO99]. Furthermore, it would also be worth evaluating alternative smoothing techniques as that proposed in [Hie00]. This smoothing technique can be interpreted as the well-known TF-IDF term weighting in information retrieval, and its application to the unigram mixture-based models presented in this chapter is an interesting open problem.

Nonetheless, the bilingual approaches described in this chapter are relatively simple models for the statistical distribution of bilingual texts. More sophisticated models, such as IBM statistical translation models [B$^+$90, B$^+$93], may be better describing the statistical distribution of bilingual, correlated texts. This latter approach is explored in Chapter 3.

Regarding multi-label text classification, we would consider alternative classifiers that directly address the multi-label problem [McC99, EW05, ZJXG05]. Although our first experience with such classifiers, represented by the BoosTexter toolkit, was rather disappointing.

Finally, the working procedure described in Section 1.2.2 can be seen as a post-processing of the output of a TC system with no actual interaction between human and computer. However a more intelligent approach would be to take advantage of the user feedback in order to refine the classification process. This process is well-studied in information retrieval and is known as relevance feedback [BYRN99]. This idea opens an appealing research line that can be explored as future work.

The bilingual models and the single-class TC results presented in this chapter were published in an international workshop:

- **J. Civera** and A. Juan. Multinomial Mixture Modelling for Bilingual Text Classification. In *Proceedings of the 6th International Workshop on Pattern Recognition in Information Systems, PRIS 2006*, pages 93–103, INSTICC Press, Paphos (Cyprus), May 2006.

The multi-label TC results of this chapter were published in an international conference:

- **J. Civera** and A. Juan. Bilingual Machine-Aided Indexing. In *Proceedings of the fifth international conference on Language Resources and Evaluation, LREC 2006*, pages 1302–1305, Genoa (Italy), May 2006.

BIBLIOGRAPHY

[ABC$^+$00] J.C. Amengual, J.M. Benedí, F. Casacuberta, A. Castaño, A. Castellanos, V. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, and J.M. Vilar. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103, 2000.

[B$^+$90] P. F. Brown et al. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

[B$^+$93] P. F. Brown et al. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of COLT '92*, pages 144–152, New York, NY, USA, 1992. ACM.

[BYRN99] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[CGGR03] N. Cancedda, E. Gaussier, C. Goutte, and J. Renders. Word-sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082, 2003.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[EC95] EC. Thesaurus eurovoc - volume 2: Subject-oriented version. Annex to the index of the Official Journal of the EC, Office for Official Publications of the EC, 1995. http://europa.eu.int/celex/eurovoc.

[EW05] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Proc. of SIGIR'05*, pages 274–281, August 2005.

[GC93] W. Gale and K. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.

[HH96] M. Hlava and R. Hainebach. Multilingual Machine Indexing. In *Proc. of NIT'96*, pages 105–121, 1996.

[Hie00] D. Hiemstra. A probabilistic justification for using tf x idf term weighting in information retrieval. *Int. J. on Digital Libraries*, 3(2):131–139, 2000.

[IO99] R. M. Iyer and M. Ostendorf. Modelling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech & Audio Processing*, 7(1):30–39, 1999.

[JV02] A. Juan and E. Vidal. On the use of Bernoulli mixture models for text classification. *Pattern Recognition*, 35(12):2705–2710, 2002.

[LD02] N. Loukachevitch and B. Dobrov. Crosslingual IR based on Multilingual Thesaurus specifically created for Automatic Text Processing. In *Proc. of SIGIR'02*, pages 105–121, August 2002.

[Lew98] D. D. Lewis. Naive Bayes at Forty: The Independence Assumption in Information Retrieval. In *Proc. of ECML'98*, pages 4–15, April 1998.

[LH00] C. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *Proc. of COLING'00*, pages 495–501, Morristown, NJ, USA, July-August 2000. Association for Computational Linguistics.

[LSST$^+$02] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[McC99] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *Proc. of AAAI'99: Workshop on Text Learning*, July 1999.

[MN98] A. McCallum and K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *Proc. of AAAI/ICML-98: Workshop on Learning for Text Categorization*, pages 41–48. Morgan Kaufmann, July 1998.

[N$^+$00] K. Nigam et al. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[NM03] J. Novovicová and A. Malík. Application of Multinomial Mixture Model to Text Classification. In F.J. Perales et al, editor, *Proc. of IbPRIA 2003*, volume LNCS 2652, pages 646–653. Lecture Notes in Computer Science, Springer-Verlag, June 2003.

[P$^+$03] B. Pouliquen et al. Automatic annotation of multilingual text collections with a conceptual thesaurus. In *Proc. of EUROLAN'03*, July-August 2003.

[P$^+$04a] D. Pavlov et al. Document Preprocessing For Naive Bayes Classification and Clustering with Mixture of Multinomials. In *Proc. of KDD'04*, pages 829–834, New York, NY, USA, August 2004. ACM.

[P$^+$04b] F. Peng et al. Augmenting Naive Bayes classifiers with statistical language models. *Information Retrieval*, 7(3):317–345, 2004.

[R$^+$03] J. Rennie et al. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proc. of ICML'03*, pages 616–623, August 2003.

[Seb02] F. Sebastiani. Machine learning in automated text categorisation. *ACM Computing Surveys*, 34(1):1–47, 2002.

[Sim98] Michel Simard. The BAF: A Corpus of English-French Bitext. In *Proc. of LREC'98*, pages 489–496, May 1998.

[SPW$^+$06] Ralf Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufis, A. Ceausu, and D. Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proc. of LREC'06*, May 2006.

[SS00] R. E. Schapire and Y. Singer. Boostexter: A boosting-based systemfor text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

[SW02] T. Scheffer and S. Wrobel. Text Classification Beyond the Bag-of-Words Representation. In *Proc. of ICML'02: Workshop on Text Learning*, pages 28–35, July 2002.

[V$^+$04] David Vilar et al. Effect of Feature Smoothing Methods in Text Classification Tasks. In *Proc. of PRIS'04*, pages 108–117, April 2004.

[V$^+$05] D. Varga et al. Parallel corpora for medium density languages. In *Proceedings of RANLP05*, pages 590–596, September 2005.

[Vap95] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[Wat00] C. Watkins. Dynamic alignment kernels. In A. J. Smola, P. L. Bartlett, B. Schlkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 39–50, Cambridge, MA, 2000. MIT Press.

[ZJXG05] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *Proc. of SIGIR'05*, pages 274–281, New York, NY, USA, August 2005. ACM.

CHAPTER 3

MIXTURE OF M1 MODELS

3.1 Introduction

In this chapter, we present a mixture extension of the M1 model, already introduced in Section 1.4, in order to define context-specific M1 models. One of the most interesting properties of mixture modelling is its capability to learn a specific probability distribution in a multimodal dataset that better explains the general data generation process. In MT these multimodal datasets are not an exception, but the general case. Indeed, it is easy to find corpora from which several topics could be drawn. These topics usually define sets of context-specific lexicons that need to be translated taking into account the semantic context in which they are found.

However, there have not been until very recently that the application of mixture modelling in statistical MT has received increasing attention. In [ZX06], three fairly sophisticated bayesian topical translation models, taking M1 model as a baseline model, were presented under the bilingual topic admixture formalism. These models capture latent topics at the document level in order to reduce semantic ambiguity and improve translation coherence. The models proposed provide in some cases better word alignment and translation quality than HMM and superior IBM models on an English-Chinese task.

In this chapter, we introduce the conventional M1 model and its EM derivation in Section 3.2, along with its mixture extension in Section 3.3. Then, two applications of the M1 model are presented: bilingual TC and statistical MT.

As regards bilingual TC, we present in Section 3.4 the M1 model in combination with the unigram language model, as an evolution of the relatively simple unigram models presented in Chapter 2. An appealing property of the unigram-M1 model is its capability to exploit the structural information contained in word correlation across languages in bilingual texts. The resultant bilingual classifier will be evaluated on the Traveller task and the BAF corpus.

As far as the application of the M1 mixture model to MT is concerned, in Section 3.5 we will focus on the Viterbi alignments obtained as a byproduct of the training process of this model. These Viterbi alignments allow us to assess the alignment and translation quality of the M1 mixture model. Finally, Section 3.6 is devoted to the conclusions of this chapter and an outlook for future work.

3.2 The M1 Model

3.2.1 The model

Following the notation introduced in Section 1.4, let $x = x_1 \ldots x_j \ldots x_{|x|}$ be a sentence in a certain source language of known length $|x|$ and let $y = y_1 \ldots y_i \ldots y_{|y|}$ be its translation of known length $|y|$ into a different target language.

For the M1 word alignment model we start from the target-conditional probability distribution $p(x \,|\, y)$, for which we define the alignment hidden variable $a = a_1 \cdots a_j \cdots a_{|x|}$, as introduced in Section 1.4. The alignment variable connects each source word to exactly one target word $a_j = \{0, \cdots, i, \cdots, |y|\}$, being 0 the position of the NULL (empty) word

$$p(x \,|\, y) = \sum_{a \in \mathcal{A}(x,y)} p(x, a \,|\, y) \tag{3.1}$$

where $\mathcal{A}(x, y)$ denotes the set of all possible alignments between x and y. Now, we can decompose the term $p(x, a \,|\, y)$ at the word-level from left to right

$$\begin{aligned} p(x, a \,|\, y) &= \prod_{j=1}^{|x|} p(x_j, a_j \,|\, x_1^{j-1}, a_1^{j-1}, y) \\ &= \prod_{j=1}^{|x|} p(a_j \,|\, x_1^{j-1}, a_1^{j-1}, y)\, p(x_j \,|\, x_1^{j-1}, a_1^{j}, y) \end{aligned} \tag{3.2}$$

where $p(a_j \,|\, x_1^{j-1}, a_1^{j-1}, y)$ is an alignment p.f. and $p(x_j \,|\, x_1^{j-1}, a_1^{j}, y)$ is a lexical p.f. or statistical dictionary. In order to define the well-known M1 model [B$^+$93], we make the following two assumptions. First, we assume that the probability of aligning a source position to a target position is uniform

$$p(a_j \,|\, x_1^{j-1}, a_1^{j-1}, y) := \frac{1}{|y| + 1}. \tag{3.3}$$

Then, we also assume that the probability of translating a source word does only depend on the target word to which is aligned

$$p(x_j \,|\, x_1^{j-1}, a_1^{j}, y) := p(x_j \,|\, y_{a_j}) \tag{3.4}$$

where $p(x_j \,|\, y_{a_j})$ is a statistical bilingual dictionary. Thus, we can rewrite Eq. (3.2) under assumptions in Eqs. (3.3) and (3.4) as

$$p(x, a \,|\, y; \boldsymbol{\Theta}) = \prod_{j=1}^{|x|} \frac{1}{|y| + 1}\, p(x_j \,|\, y_{a_j}) \tag{3.5}$$

where

$$\mathbf{\Theta} = \{\ p(u \mid v) \quad u \in \mathcal{X}, v \in \mathcal{Y}\ \} \tag{3.6}$$

is a statistical bilingual dictionary.

The model using indicator vectors

As we did in Chapter 2, we change the nature of the original alignment variable $a_j \in \{0, \ldots, |y|\}$, from an integer value into an indicator vector

$$\boldsymbol{a}_j = (a_{j0}, a_{j1}, \ldots, a_{j|y|})^t. \tag{3.7}$$

The vector $\boldsymbol{a}_j$ values one in the ith position and zeros elsewhere, if the source position j is aligned to the target position i. Equivalently to Eq. (3.5), we have

$$p(x, a \mid y; \mathbf{\Theta}) = \prod_{j=1}^{|x|} \prod_{i=0}^{|y|} \left[\frac{1}{|y|+1}\, p(x_j \mid y_i) \right]^{a_{ji}}. \tag{3.8}$$

According to this notation, the initial model in Eq. (3.1) can be rewritten as follows

$$\begin{aligned}
p(x \mid y; \mathbf{\Theta}) &= \sum_{a} p(x, a \mid y; \mathbf{\Theta}) \\
&= \frac{1}{(|y|+1)^{|x|}} \sum_{\boldsymbol{a}_1} \cdots \sum_{\boldsymbol{a}_{|x|}} \prod_{j=1}^{|x|} \prod_{i=0}^{|y|} p(x_j \mid y_i)^{a_{ji}} \\
&= \frac{1}{(|y|+1)^{|x|}} \sum_{\boldsymbol{a}_1} \prod_{i=0}^{|y|} p(x_1 \mid y_i)^{a_{1i}} \left[\sum_{\boldsymbol{a}_2} \cdots \sum_{\boldsymbol{a}_{|x|}} \prod_{j=2}^{|x|} \prod_{i=0}^{|y|} p(x_j \mid y_i)^{a_{ji}} \right] \\
&= \frac{1}{(|y|+1)^{|x|}} \prod_{j=1}^{|x|} \sum_{\boldsymbol{a}_j} \prod_{i=0}^{|y|} p(x_j \mid y_i)^{a_{ji}} \\
&= \prod_{j=1}^{|x|} \sum_{i=0}^{|y|} \frac{1}{|y|+1}\, p(x_j \mid y_i).
\end{aligned} \tag{3.9}$$

Eq. (3.9) is the usual form of the M1 model that only depends on a bilingual dictionary. The M1 model makes the naive assumption that source words are conditionally independent given y

$$p(x \mid y; \mathbf{\Theta}) = \prod_{j=1}^{|x|} p(x_j \mid y) \tag{3.10}$$

where

$$p(x_j \mid y) = \sum_{i=0}^{|y|} \frac{1}{|y|+1}\, p(x_j \mid y_i) \tag{3.11}$$

is the average probability of x_j to be translated into a target word in y.

3.2.2 Maximum likelihood estimation

In this section we derive an EM algorithm for the maximum likelihood estimation of $\mathbf{\Theta}$ in the M1 model w.r.t. a set of independent samples. Let $(X, Y) = ((x_1, y_1), \ldots, (x_n, y_n), \ldots, (x_N, y_N))^t$ be N samples independently drawn according to the probability distribution defined by an M1 model of parameters $\mathbf{\Theta}$, being $x_n = \left(x_{n1}, \ldots, x_{nj}, \ldots, x_{n|x|}\right)$ and $y_n = \left(y_{n1}, \ldots, y_{ni}, \ldots, y_{n|y|}\right)$ the sequence of source and target words of the nth sample. The log-likelihood function of $\mathbf{\Theta}$ is

$$L(\mathbf{\Theta}; X, Y) = \sum_{n=1}^{N} \sum_{j=1}^{|x_n|} \log \sum_{i=0}^{|y_n|} \frac{1}{|y_n| + 1}\, p(x_{nj} \,|\, y_{ni}). \tag{3.12}$$

Now, let A be the set of alignment indicator vectors associated with the bilingual pairs (X, Y) with

$$A = (a_1, \ldots, a_n, \ldots, a_N)^t\,. \tag{3.13}$$

The variable A is the missing data in the M1 model, playing the role of Z in Section 1.8. As in Eq. (1.27), the E step computes the expected value of the logarithm of $p(X, A \,|\, Y)$, given the (incomplete) data samples (X, Y) and a current estimate of $\mathbf{\Theta}$, $\mathbf{\Theta}^{(k)}$. Given that the alignment variables in A are independent from each other, we can compute the E step equivalently to Eq. (1.44),

$$Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) = \sum_{n=1}^{N} \sum_{j=1}^{|x_n|} \sum_{i=0}^{|y_n|} a_{nji}^{(k)} \left[\log \frac{1}{|y_n| + 1} + \log p(x_{nj} \,|\, y_{ni})\right] \tag{3.14}$$

with

$$a_{nji}^{(k)} = \frac{p(x_{nj} \,|\, y_{ni})^{(k)}}{\sum_{i'=0}^{|y_n|} p(x_{nj} \,|\, y_{ni'})^{(k)}}. \tag{3.15}$$

That is, the expectation of word x_{nj} to be connected to y_{ni} is our current estimation of the probability of x_{nj} to be translated into y_{ni}, instead of any other word in y_n (including the NULL word).

In the M step, we maximise Eq. (3.14), as seen in Section 1.8, in order to obtain the standard update formula for the M1 model,

$$p(u \,|\, v)^{(k+1)} = \frac{N(u, v)}{\sum\limits_{u' \in \mathcal{X}} N(u', v)} \qquad \forall u \in \mathcal{X}, v \in \mathcal{Y} \tag{3.16}$$

where

$$N(u, v) = \sum_{n=1}^{N} \sum_{j=1}^{|x_n|} \sum_{i=0}^{|y_n|} \delta(x_{nj} = u)\, \delta(y_{ni} = v)\, a_{nji}^{(k)}\,. \tag{3.17}$$

The estimation of $p(u \,|\, v)$ can be seen as a normalised partial count of how many times the source word u is aligned to the target word v.

3.3 Mixture of M1 models

3.3.1 The model

Eq. (3.9) is a relatively simple parametric model for distributions of bilingual pairs of sentences. Then, it is a good choice to describe simple distributions, but it might not be so good to approximate complex distributions, such as those comprising many topically-unrelated groups of pairs. To deal with such cases, we will use the idea of finite mixture modelling and replace our simple model in Eq. (3.9) by a y conditional finite mixture.

Considering that our bilingual pairs are drawn from different topics (contexts), we can rewrite the conditional translation p.f. $p(x \mid y)$ as a y-conditional finite mixture, similarly to Eq. (1.41),

$$p(x \mid y; \boldsymbol{\Theta}) = \sum_{\boldsymbol{z}} \prod_{t=1}^{T} [p(t)\, p(x \mid y, t; \boldsymbol{\Theta}_t)]^{z_t} \tag{3.18}$$

where we have implicitly assumed that $\boldsymbol{z}$ does not depend on y when modelling $p(t)$. On the other hand, $p(x \mid y, t; \boldsymbol{\Theta}_t)$ is a component-dependent version of an alignment translation model,

$$p(x \mid y, t; \boldsymbol{\Theta}_t) = \sum_{a} p(x, a \mid y, t; \boldsymbol{\Theta}_t). \tag{3.19}$$

playing the role of the component-conditional p.f. in Eq. (1.41). Now, we can plug Eq. (3.19) into Eq. (3.18) and reorganise the resulting expression

$$\begin{aligned} p(x \mid y; \boldsymbol{\Theta}) &= \sum_{\boldsymbol{z}} \prod_{t=1}^{T} \left[p(t) \sum_{a} p(x, a \mid y, t; \boldsymbol{\Theta}_t) \right]^{z_t} \\ &= \sum_{t=1}^{T} p(t) \sum_{a} p(x, a \mid y, t; \boldsymbol{\Theta}_t) \\ &= \sum_{a} \sum_{t=1}^{T} p(t)\, p(x, a \mid y, t; \boldsymbol{\Theta}_t) \\ &= \sum_{\boldsymbol{z}} \sum_{a} \prod_{t=1}^{T} [p(t)\, p(x, a \mid y, t; \boldsymbol{\Theta}_t)]^{z_t} \end{aligned} \tag{3.20}$$

to ease the presentation of the EM algorithm in the current and subsequent translation mixture models in Chapters 4 and 5.

In the *M1 mixture model*, the component-conditional p.f. $p(x, a \mid y, t; \boldsymbol{\Theta}_t)$ in Eq. (3.20) becomes a component-dependent version of Eq. (3.8),

$$p(x, a \mid y, t; \boldsymbol{\Theta}_t) = \prod_{j=1}^{|x|} \prod_{i=0}^{|y|} \left[\frac{1}{|y| + 1}\, p(x_j \mid y_i, t) \right]^{a_{ji}} \tag{3.21}$$

where $\boldsymbol{\Theta}_t$ is a component-dependent dictionary

$$\boldsymbol{\Theta}_t = \left\{ \; p(u \,|\, v, t) \quad u \in \mathcal{X}, v \in \mathcal{Y} \; \right\} . \tag{3.22}$$

3.3.2 Maximum likelihood estimation

The log-likelihood function of $\boldsymbol{\Theta}$ w.r.t. N independent samples is

$$L(\boldsymbol{\Theta}; X, Y) = \sum_{n=1}^{N} \log \sum_{\boldsymbol{z}_n} \sum_{a_n} \prod_{t=1}^{T} \left[p(t)\, p(x_n, a_n \,|\, y_n, t; \boldsymbol{\Theta}_t) \right]^{z_{nt}} . \tag{3.23}$$

For the application of the EM algorithm, we consider Z and A, as defined in Eqs. (1.43) and (3.13) respectively, to be the missing data Z in Eq (1.27). Thus, equivalently to Eq. (1.44), the function Q becomes

$$\begin{aligned} Q(\boldsymbol{\Theta} \,|\, \boldsymbol{\Theta}^{(k)}) = & \sum_{n=1}^{N} \sum_{t=1}^{T} z_{nt}^{(k)} \log p(t) \\ & + \sum_{j=1}^{|x_n|} \sum_{i=0}^{|y_n|} (z_{nt}\, a_{nji})^{(k)} \left[\log \frac{1}{|y_n| + 1} + \log p(x_{nj} \,|\, y_{ni}, t) \right] . \end{aligned} \tag{3.24}$$

So, the E step requires the calculation of $z_{nt}^{(k)}$ and $(z_{nt}\, a_{nji})^{(k)}$. The computation of $z_{nt}^{(k)}$ is similar to that of Eq. (1.45), substituting the component-conditional p.f. by a component-conditional M1 p.f., that is,

$$z_{nt}^{(k)} = \frac{p(t)^{(k)}\, p(x_n \,|\, y_n, t; \boldsymbol{\Theta}_t^{(k)})}{\sum_{t'=1}^{T} p(t')^{(k)}\, p(x_n \,|\, y_n, t'; \boldsymbol{\Theta}_{t'}^{(k)})} \tag{3.25}$$

where $z_{nt}^{(k)}$ is the posterior probability of the tth-component having generated the nth sample (x_n, y_n). Regarding $(z_{nt}\, a_{nji})^{(k)}$, we have,

$$\begin{aligned} (z_{nt}\, a_{nji})^{(k)} &= p(z_{nt} = 1, a_{nji} = 1 \,|\, x_n, y_n) \\ &= p(z_{nt} = 1 \,|\, x_n, y_n)\, p(a_{nji} = 1 \,|\, z_{nt} = 1, x_n, y_n) \\ &= z_{nt}^{(k)}\, a_{njit}^{(k)} \end{aligned} \tag{3.26}$$

with $a_{njit}^{(k)}$ being a component-dependent version of $a_{nji}^{(k)}$ in Eq. (3.15),

$$a_{njit}^{(k)} = \frac{p(x_{nj} \,|\, y_{ni}, t)^{(k)}}{\sum_{i'=0}^{|y_n|} p(x_{nj} \,|\, y_{ni'}, t)^{(k)}} \tag{3.27}$$

where $a_{njit}^{(k)}$ can be thought of the posterior probability of the source position j to be aligned to the target position i in the tth component for the nth sample (x_n, y_n).

In the M step, we maximise Eq. (3.24) to obtain a new set of parameters $\mathbf{\Theta}^{(k+1)}$. The new component priors are computed as in Eq. (1.48), and the update equation for the component-dependent dictionaries is

$$p(u \,|\, v, t)^{(k+1)} = \frac{N(u, v, t)}{\sum\limits_{u' \in \mathcal{X}} N(u', v, t)} \qquad \forall t, u \in \mathcal{X}, v \in \mathcal{Y} \tag{3.28}$$

where

$$N(u, v, t) = \sum_{n=1}^{N} z_{nt}^{(k)} \sum_{j=1}^{|x_n|} \sum_{i=0}^{|y_n|} \delta(x_{nj} = u)\, \delta(y_{ni} = v)\, a_{njit}^{(k)}\,. \tag{3.29}$$

The estimation of $p(u \,|\, v, t)$ can be understood as a normalised partial count of how many times the source word u is aligned to the target word v, weighted by the posterior probability (responsibility) of the tth component having generated the nth sample. The asymptotic cost of the training algorithm per iteration is $O(N \cdot T \cdot \overline{|x|} \cdot \overline{|y|})$, where $\overline{|x|}$ and $\overline{|y|}$ are the source and target average lengths, respectively.

3.3.3 Smoothing

The component-conditional dictionary is smoothed at two levels to avoid overfitting problems and zero probabilities for rare words. On the one hand, we smooth the estimated statistical dictionary interpolating with a uniform distribution over the source vocabulary

$$\hat{p}(u \,|\, v, t) = (1 - \epsilon)\, p(u \,|\, v, t) + \epsilon\, \frac{1}{|\mathcal{X}|}. \tag{3.30}$$

This smoothing technique follows the same idea that we presented in Section 2.3.3 to smooth the unigram distribution and so, the ϵ parameter was manually set in order to obtain smoothed error and log-likelihood curves as we increase the number of components in the mixture model.

On the other hand, we smooth the component-conditional statistical dictionary (specific distribution) interpolating with the conventional statistical dictionary (general distribution)

$$\hat{p}(u \,|\, v, t) = \left[1 - \frac{\alpha}{\alpha + p(v)}\right] p(u \,|\, v, t) + \frac{\alpha}{\alpha + p(v)}\, p(u \,|\, v). \tag{3.31}$$

The interpolation coefficient depends on the interpolation parameter α, and on the unigram probability of the target word v, that is, the relative frequency of the target word v on the training set.

This interpolation parameter α defines the target-word unigram probability threshold at which the specific and general distribution are equally weighted. For those target words above this threshold (higher unigram probability), the specific distribution will dominate, while for those target words below the threshold (lower unigram probability), the general distribution dominates. See Figure 3.1 to observe the evolution of the interpolation coefficient, as a function of the unigram probability of the target word for a given $\alpha = 1e - 3$. It should be noticed that the interpolation coefficient is equal to 0.5 when $p(v) = \alpha$.

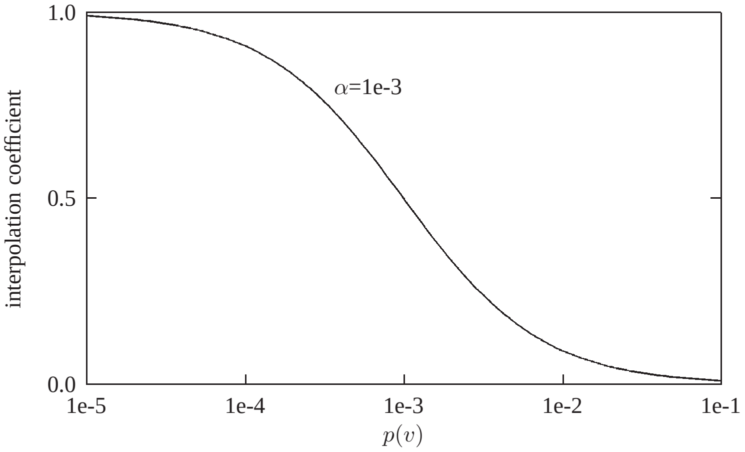


Figure 3.1: Interpolation coefficient curve with interpolation parameter $\alpha = 1e - 3$ as a function of the unigram probability of the target word. It should be noticed that the interpolation coefficient is equal to 0.5 when $p(v) = \alpha$.

Intuitively, the interpolation parameter is a powerful way to control at which relative frequency is worth considering a component-dependent statistical dictionary for a target word. The idea behind this smoothing technique is that, target words with high frequency may have different translations depending on the context and, given their high frequency on the training set, their context-specific dictionary can be correctly estimated. On the contrary, low frequency target words might have fewer translations, and in any case, their corresponding dictionary cannot be adequately estimated due to their few occurrences.

During the EM training of the M1 mixture model, smoothing is applied at the end of each M step. Besides, the parameter α is manually tuned to optimise the evaluation metric in question on the development set. This smoothing technique is inspired on that of the fertility distribution presented in [ON03].

3.4 Bilingual text classification using the M1 model

In this section, we present the application of the M1 model to bilingual TC. Our goal is to study the contribution of cross-lingual structural information in order to improve the accuracy of bilingual text classifiers. To this purpose, we combine the unigram and M1 models under the finite mixture modelling. As we did in Chapter 2, we first derive the model and its maximum likelihood estimation, followed by the view of this model as a bilingual text classifier and its maximum likelihood estimation in the framework of supervised classification.

3.4.1 Mixture of unigram-M1 models

The model

Let (x, y) be a bilingual pair of source-target sentences coming from a T-component mixture model, equivalently to Eq. (1.41),

$$p(x, y; \boldsymbol{\Theta}) = \sum_{\boldsymbol{z}} \prod_{t=1}^{T} [p(t)\, p(y \mid t; \boldsymbol{\Theta}_t)\, p(x \mid y, t; \boldsymbol{\Theta}_t)]^{z_t} \tag{3.32}$$

where, for the *unigram-M1* mixture model, $p(t)$ is its mixture coefficient as in Eq. (1.39), $p(y \mid t; \boldsymbol{\Theta}_t)$ is a component-conditional unigram model as in Eq (2.5) and $p(x \mid y, t; \boldsymbol{\Theta}_t)$ is a component-conditional M1 model as in Eq (3.19). The parameter vector $\boldsymbol{\Theta}$ has the usual form of a mixture model in Eq. (1.34), and each component has its own vector of parameters

$$\boldsymbol{\Theta}_t = \begin{cases} p(v \mid t) & v \in \mathcal{Y} \\ p(u \mid v, t) & u \in \mathcal{X}, v \in \mathcal{Y} \end{cases} . \tag{3.33}$$

It is important to note the substantial difference between the bilingual unigram mixture model in Eq. (2.13) and the *unigram-M1* mixture model in Eq. (3.32), in which x depends on y as a result of the inclusion of a translation model.

Maximum likelihood estimation

As in previous models, we use the EM algorithm to compute a maximum likelihood estimation of $\boldsymbol{\Theta}$ w.r.t. N independent samples $(X, Y) = ((x_1, y_1), \ldots, (x_N, y_N))^t$. The log-likelihood function of $\boldsymbol{\Theta}$ is

$$L(\boldsymbol{\Theta}; X, Y) = \sum_{n=1}^{N} \log \sum_{\boldsymbol{z}_n} \prod_{t=1}^{T} [p(t)\, p(y_n \mid t; \boldsymbol{\Theta}_t)\, p(x_n \mid y_n, t; \boldsymbol{\Theta}_t)]^{z_{nt}} \tag{3.34}$$

and considering Z and A to be the missing data in Eq. (1.27), we have the following Q function

$$\begin{aligned} Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)}) = & \sum_{n=1}^{N} \sum_{t=1}^{T} z_{nt}^{(k)} \log p(t) + \sum_{i=1}^{|y_n|} z_{nt}^{(k)} \log p(y_{ni} \mid t) \\ & + \sum_{j=1}^{|x_n|} \sum_{i'=0}^{|y_n|} (z_{nt}\, a_{nji'})^{(k)} \left[\log \frac{1}{|y_n| + 1} + \log p(x_{nj} \mid y_{ni'}, t) \right] \end{aligned} \tag{3.35}$$

where we need to calculate the expected value of z_{nt} and $z_{nt}\, a_{nji'}$, as in Section 3.3.2. However, the computation of $z_{nt}^{(k)}$ differs from that in Eq. (1.45) in the underlying component-conditional p.f.,

$$z_{nt}^{(k)} = \frac{p(t)^{(k)}\, p(x_n, y_n \,|\, t; \boldsymbol{\Theta}_t^{(k)})}{\sum_{t'=1}^{T} p(t')^{(k)}\, p(x_n, y_n \,|\, t'; \boldsymbol{\Theta}_{t'}^{(k)})} \tag{3.36}$$

where $z_{nt}^{(k)}$ is the posterior probability of (x_n, y_n) being actually generated by the tth component of a unigram-M1 mixture model. On the other hand, the decomposition of $(z_{nt}\, a_{nji'})^{(k)}$ is analogous to that of Eq. (3.26), where $a_{nji't}^{(k)}$ is computed as in Eq. (3.27).

In the M step, we obtain an updated set of parameters $\boldsymbol{\Theta}^{(k+1)}$. The component priors are computed as in Eq. (1.48), the component-conditional unigram language model are updated according to Eq. (2.19), and an update equation for the component-conditional M1 model is given in Eq. (3.28).

3.4.2 Bilingual text classification using the unigram-M1 model

The decision rule

As we did in Section 2.4.1 for the unigram and bilingual unigram models, the unigram-M1 model can be also used as a class-conditional model in supervised bilingual TC tasks. Here, the Bayes' rule for the unigram-M1 mixture model is

$$c(x, y) = \arg\max_{c}\ \log p(c) + \log p(x, y \,|\, c) \tag{3.37}$$

where

$$p(x, y \,|\, c) = \sum_{t=1}^{T} p(t \,|\, c) \prod_{i=1}^{|y|} p(y_i \,|\, t, c)\ \prod_{j=1}^{|x|} \sum_{i'=0}^{|y|} \frac{1}{|y|+1}\, p(x_j \,|\, y_{i'}, t, c) \tag{3.38}$$

Maximum likelihood estimation for supervised classification

As in Section 2.4.2, we extend the single supervised class training presented in Section 3.4.1 to train several supervised classes at the same time.

Let $(X, Y, C) = ((x_1, y_1, c_1), \ldots, (x_N, y_N, c_N))^t$ be the set of training samples, and let $\boldsymbol{\Psi}$ be the vector of unknown parameters as defined in Eq. (2.27). where the class-conditional p.f. is a unigram-M1 mixture model controlled by a vector $\boldsymbol{\Theta}_c$, as defined in Section 3.4.1 for $\boldsymbol{\Theta}$. The log-likelihood of $\boldsymbol{\Psi}$ w.r.t. the labelled data is, equivalently to Eq. (2.28),

$$L(\boldsymbol{\Psi}; X, Y, C) = \sum_{c=1}^{C} N_c\ \log p(c) + L_c(\boldsymbol{\Theta}_c; X_c, Y_c) \tag{3.39}$$

where Y_c is defined analogously to X_c, and

$$L_c(\mathbf{\Theta}_c; X_c, Y_c) = \sum_{n=1}^{N} \delta(c_n = c) \log \sum_{\boldsymbol{z}_n} \prod_{t=1}^{T} \left[p(t \mid c_n)\ p(x_n, y_n \mid t, c_n; \mathbf{\Theta}_{c_n t})\right]^{z_{nt}} \tag{3.40}$$

which is optimised by extending the EM algorithm presented in Section 3.4.1.

The E step computes Eqs. (3.36) and (3.27) using $\mathbf{\Theta}_{c_n}$ for those training samples of the form (x_n, y_n, c_n). So we have

$$z_{nt}^{(k)} = \frac{p(t \mid c_n)^{(k)}\, p(y_n \mid t, c_n)^{(k)}\, p(x_n \mid y_n, t, c_n)^{(k)}}{\sum_{t'=1}^{T} p(t' \mid c_n)^{(k)}\ p(y_n \mid t', c_n)^{(k)}\, p(x_n \mid y_n, t', c_n)^{(k)}} \tag{3.41}$$

and

$$a_{njit}^{(k)} = \frac{p(x_{nj} \mid y_{ni}, t, c_n)^{(k)}}{\sum_{i'=0}^{|y_n|} p(x_{nj} \mid y_{ni'}, t, c_n)^{(k)}}. \tag{3.42}$$

The M step computes the new set of parameters $\mathbf{\Psi}^{(k+1)}$. More precisely, we calculate class priors as in Eq. (2.31), class-conditional mixture coefficients as in Eq. (2.32), class-conditional unigram parameters as in Eq. (2.33) and class-conditional statistical dictionaries as

$$p(u \mid v, t, c)^{(k+1)} = \frac{N(u, v, t, c)}{\sum\limits_{u' \in \mathcal{X}} N(u', v, t, c)} \qquad \forall c, t, u \in \mathcal{X}, v \in \mathcal{Y} \tag{3.43}$$

where

$$N(u, v, t, c) = \sum_{n=1}^{N} \delta(c_n = c)\, z_{nt}^{(k)} \sum_{j=1}^{|x_n|} \sum_{i=0}^{|y_n|} \delta(x_{nj} = u)\, \delta(y_{ni} = v)\, a_{njit}^{(k)}\,. \tag{3.44}$$

3.4.3 Experimental results

The unigram-M1 mixture model described in the previous section was assessed on the two tasks described in Chapter 2: the Traveller dataset and the BAF corpus.

Several experiments were carried out to analyse the behaviour of the unigram-M1 classifier in terms of log-likelihood and classification error rate as a function of the number of mixture components per class ($T \in \{1, 2, 5, 10, 20, 50, 100\}$). These experiments were carried out on the same training and test partitions defined in Chapter 2 for Traveller and BAF.

Figures 3.2 and 3.3 shows the evolution of the error rate (left y axis) and log-likelihood (right y axis), on the training and test sets of the Traveller and BAF, respectively, for an increasing number of mixture components (x axis). Each plotted point is an average over values obtained from 30 randomised trials.

From the results in Figures 3.2 and 3.3, we can see that the evolution of the log-likelihood on the training set is as theoretically expected, in both Traveller

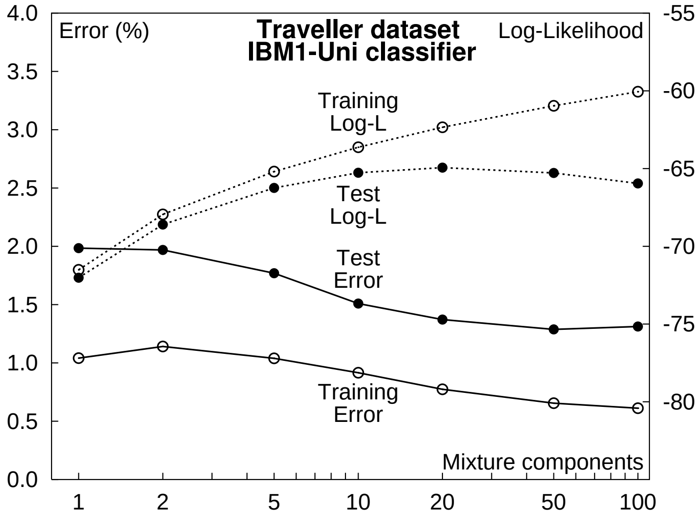


Figure 3.2: Error rate and log-likelihood curves in training and test sets as a function of the number of mixture components, in Traveller for the unigram-M1 classifier.

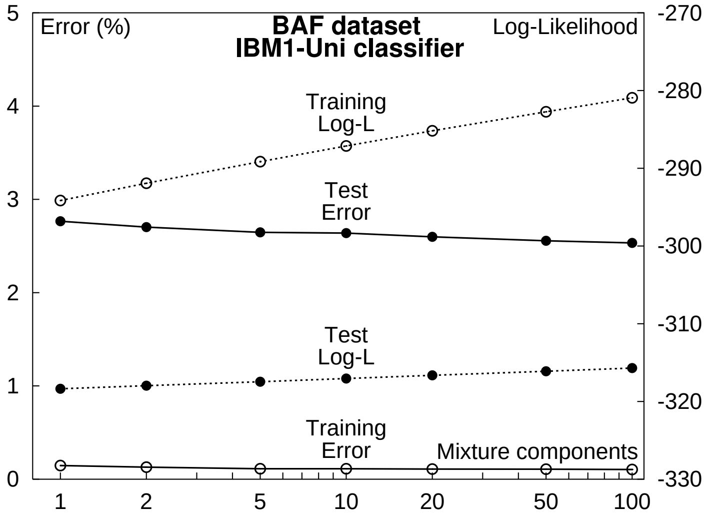


Figure 3.3: Error rate and log-likelihood curves in training and test sets as a function of the number of mixture components, in BAF for the unigram-M1 classifier.

and BAF. In the test set of the Traveller task, the log-likelihood increases up to a moderate number of components 20, while the best error rate is obtained with 50 components.

Table 3.1: Summary error table of monolingual and bilingual classifiers on the Traveller task and the BAF corpus.

	mix 1g	mix 1g1g	mix 1gM1	SVM^{light}	BoosTexter
Traveller	1.5	1.4	1.3	1.5	1.2
BAF	4.1	3.0	2.5	9.0	5.8

However, in the test partition of the BAF corpus, the log-likelihood keeps increasing (and the error rate decreasing) even after 100 components per mixture. This uncommon behaviour may be explained in the light of the statistics of the BAF corpus where one third of the words occurs only once in the corpus as a whole and even less than that in some of the classes. This data scarcity feature makes very difficult for a model such as M1 to learn word correlations across languages resulting in an expected overfitting effect.

Figure 3.4 compares the performance of the best monolingual (English-based), the bilingual local, both of them presented in Chapter 2, and the unigram-M1 classifiers. As shown, the unigram-M1 classifier outperforms the monolingual and bilingual local classifiers, but the difference is not so important in the Traveller as in the BAF corpus. Therefore, the word correlation across languages that provides the M1 model helps to improve the accuracy of its classifier.

Table 3.1 presents a summary of the error figures of the different classifiers on the Traveller task and the BAF corpus. As we can observe, the unigram-M1 (mix 1gM1) mixture model supersedes the other two unigram models, being statistically significant better in the case of the BAF corpus, but not being so for the Traveller task. The unigram-M1 mixture model obtains similar performance to SVM and boosting methods in the Traveller task, and statistically significantly better in the BAF corpus. These experiments show the benefits of learning word correlation across languages in bilingual TC.

3.5 Mixture of M1 models applied to MT

In this section, we describe the computation of the Viterbi alignments for the M1 mixture model and then, we directly and indirectly evaluate this model on well-known MT tasks.

First, we decided to carry out the direct evaluation in terms of alignment error rate (AER) using the Hansard shared task. Although it is unclear the relation between alignment quality (AER) and translation quality (BLEU), if any [AD06, FM07b]. Being that as it is, this measure is still a useful instrument to directly gauge the quality of novel models as its ability to map source to target positions [ZX06, FM07a].

Secondly, we indirectly assess the translation quality of the proposed model by training a phrase-based system from its Viterbi alignments. We are aware that this evaluation procedure of the models may mask their actual performance, but it

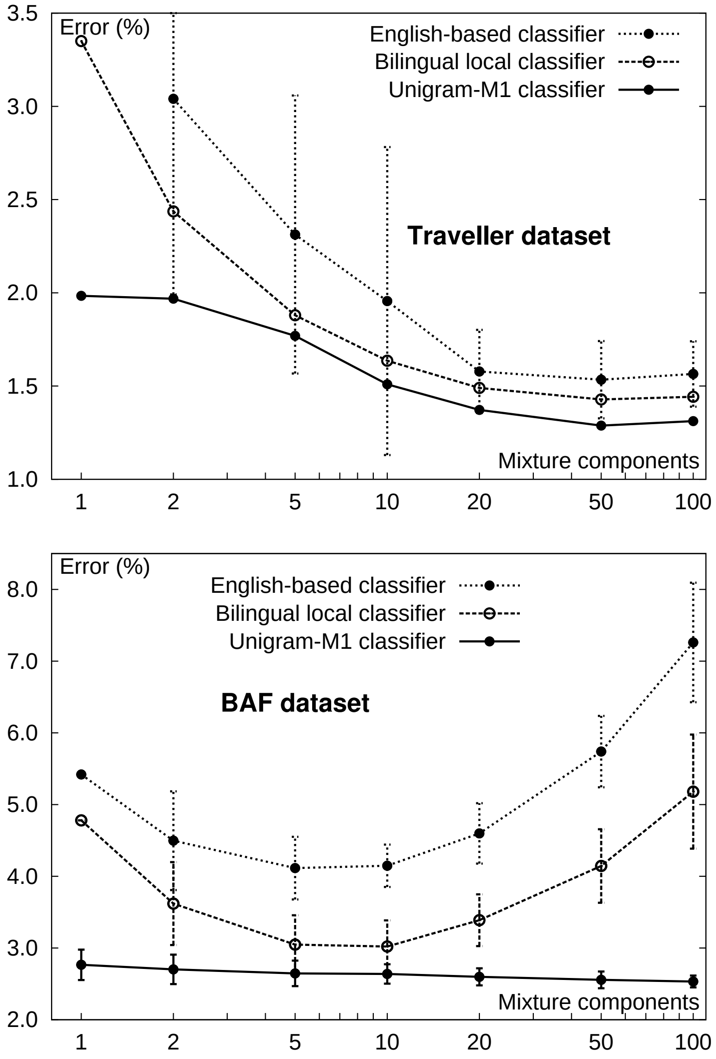


Figure 3.4: Competing curves: %error vs. mixture components for Traveller and BAF.

allows us to compare the translation quality of the proposed models to other translation system trained in a similar fashion. Moreover, we can analyse the evolution of the translation quality of the system, in terms of BLEU score, as a function of the number of components in the mixture.

3.5.1 Viterbi alignment

In Eq. (3.1) we introduced the concept of alignment as an assignment between source and target words, more precisely between source and target positions. However, this alignment information was missing in the translation process, and we had to marginalise over all possible values of the alignment variable.

In practise, we are interested in the most probable alignment, also known as the Viterbi alignment, given the source and target sentences and an estimate of the model parameters,

$$\hat{a} = \arg\max_{a} p(a \mid x, y; \mathbf{\Theta}) \tag{3.45}$$

that, considering that we are maximising over a, can be easily rewritten as

$$\hat{a} = \arg\max_{a} p(x, a \mid y; \mathbf{\Theta}). \tag{3.46}$$

Assuming a conventional M1 model, Eq. (3.46) can be transformed into

$$\begin{aligned}\hat{a} &= \arg\max_{a} \prod_{j=1}^{|x|} \frac{1}{|y|+1}\, p(x_j \mid y_{a_j}) \\ &= \arg\max_{a} \prod_{j=1}^{|x|} p(x_j \mid y_{a_j}) \end{aligned} \tag{3.47}$$

whose maximisation is trivial

$$\hat{a} = \hat{a}_1, \dots, \hat{a}_j, \dots, \hat{a}_{|x|} \tag{3.48}$$

with

$$\hat{a}_j = \max_{a_j} p(x_j \mid y_{a_j}).$$

In other words, the Viterbi alignment for the M1 model is computed as a local maximisation for each source position, being its asymptotic cost $O(|x| \cdot |y|)$.

Nevertheless, the computation of the Viterbi alignment for the M1 mixture model

$$\hat{a} = \arg\max_{a} \sum_{t=1}^{T} p(t) \prod_{j=1}^{|x|} \frac{1}{|y|+1}\, p(x_j \mid y_{a_j}, t) \tag{3.49}$$

is approximated by maximising over the components in the mixture,

$$\hat{a} \approx \arg\max_{a} \max_{t=1,\dots,T}\; p(t) \prod_{j=1}^{|x|} p(x_j \mid y_{a_j}, t) \tag{3.50}$$

being its asymptotic cost $O(T \cdot |x| \cdot |y|)$.

3.5.2 Evaluation of alignment quality

Corpora

The corpus employed in the experiments was the French-English Hansard task consisting on the debates of the Canadian parliament. This corpus is one of the

Table 3.2: Statistics on the French-English Hansard task (K denotes $\times 1.000$, and M denotes $\times 1.000.000$)

	Training set		Trial set		Test set	
	Fr	En	Fr	En	Fr	En
sent. pairs	1.1M		37		447	
average length	20	17	19	17	17	15
vocabulary size	87K	68K	344	322	1943	1732
running words	24M	20M	721	661	7761	7020
singletons	27K	20K	265	238	1323	1103

resources that were used during the word alignment shared task organised at the HLT/NAACL 2003 workshop on ”Building and Using Parallel Texts”. See statistics in Table 3.2.

The measures defined above are computed on an independent test set randomly drawn that was manually labelled by two annotators. Each annotator comes up with a S and P alignment set. The S alignment sets from each annotator are intersected to defined the reference S alignment set, while the reference P alignment set is the result of the union of the P alignment sets from both annotators. The definition of the S and P alignment sets in this ways guarantees an alignment error rate of zero percent when we compare the S alignments of each annotator with the reference alignment.

The training partition was filtered according to the GIZA++ standards to ease the comparison with this toolkit, that is, sentences whose length is above 100 words were truncated and those sentence pairs whose ratio between the source and target length is more than 9 were shortened to the minimum of the source and target length.

Experimental results

The objective of these experiments is to study the evolution of AER as a function of the number of components in the M1 mixture model on the Hansard task. The results reported with the GIZA++ toolkit are mostly for sanity check reasons. For this reason, this kind of experiments were not carried out for the evaluation of translation quality in Section 3.5.3. The smoothing parameters were manually tuned on the trial partition to minimise AER.

Table 3.3 presents AER figures on the test partition for the M1 mixture model. Each number in Table 3.3 is an average over values obtained from 10 randomised initialisation. These experiments were performed for both directions, English-French (En-Fr) and French-English (Fr-En) and varying the number of components in the mixture model ($T = 1, 2, 3$). The training scheme (number of iterations per model) was $mix1^5$. The computation of the Viterbi alignments was calculated according to Eq. (3.50). As observed in Table 3.3, it does not seem to be a clear

Table 3.3: AER figures on the test partition of the Hansard corpus for the M1 mixture model varying the number of components in the mixture ($T = 1, 2, 3$) and the conventional M1 model implemented in the GIZA++ toolkit.

	GIZA++	1	2	3
Fr-En	27.8	27.3	27.3	27.4
En-Fr	24.3	24.4	24.4	24.4

contribution by applying mixture modelisation to the M1 model on either of the two directions.

3.5.3 Evaluation of translation quality

Corpora

The dataset that was used in the experiments for the M1 mixture model was obtained from the shared-task of the ACL 2007 statistical MT workshop on “Machine Translation for European Languages” [CB$^+$07]. This dataset includes four partitions devoted to different purposes:

- Training sets for translation models.
- Development sets to tune translation systems (see statistics in Tables 3.4 and 3.5).
- Test development sets to evaluate translation systems (see statistics in Tables 3.4 and 3.5).
- Monolingual training sets for language models (see statistics in Table 3.6).

The first three partitions include data coming from both corpora, Europarl and News-Commentary, however the last partition only includes data coming from the Europarl corpus. It would be possible to enrich the latter partition with data from other training partitions [KS07], but we decided not to do so since our focus is the study of context-specific translation models.

The shared task described above is composed of two corpora, the Europarl version 3 and the News-Commentary, although the number of sentences of the latter constitutes less than 5% of the number of sentences of the former. It is important to remark that the domain of the Europarl and News-Commentary corpora is different, and this is an interesting characteristic that our mixture model can exploit in order to learn domain-specific translation models. To this purpose, we concatenated the training sets for translation models of the Europarl and the News-Commentary corpora (see statistics in Table 3.7), letting the mixture model distinguish which sentence pairs should contribute to learn a given M1 component in the mixture.

Table 3.4: Statistics of the English, Spanish, French, German development and test partitions for the Europarl corpus (K denotes $\times 1.000$, and M denotes $\times 1.000.000$).

Europarl	development set				test set			
	En	Es	Fr	De	En	Es	Fr	De
sentences (K)	2	2	2	2	2	2	2	2
average length (words)	29	30	32	28	29	30	32	27
running words (Kwords)	59	61	64	55	58	60	63	54
perplexity	74	75	63	118	72	76	63	117

Table 3.5: Statistics of the English, Spanish, French, German development and test partitions for the News-Commentary corpus (K denotes $\times 1.000$, and M denotes $\times 1.000.000$).

News-Commentary	development set				test set			
	En	Es	Fr	De	En	Es	Fr	De
#sentences (K)	1	1	1	1	1	1	1	1
average length (words)	24	28	29	25	24	28	29	25
running words (Kwords)	26	29	31	26	26	30	31	27
perplexity	225	155	120	322	248	164	134	339

The language pairs involved in the experiments were {Spanish,French,German}-English. Both corpora were preprocessed as suggested for the baseline system by tokenising, filtering sentences longer than 40 words and lowercasing. This same corpora will be employed in the evaluation of the M2 and HMM mixture models (see Chapters 4 and 5, respectively).

Regarding the statistics of Table 3.7, it should be noticed the ratio between vocabulary size and singletons (words that occur only once) ranging from 35% in English to 50% in German, indicates somehow the complexity of the task, while the number of sentence pairs and running words give a clear idea of the magnitude

Table 3.6: Statistics of the English, Spanish, French, German monolingual training partitions used to train language models (K denotes $\times 1.000$, and M denotes $\times 1.000.000$).

Europarl	En	Es	Fr	De
sentences (M)	1.4	1.4	1.4	1.5
average length (words)	27	28	30	25
running words (Mwords)	38	40	43	37

Table 3.7: Statistics of the concatenation of the $\{$Spanish,French,German$\}$-English training partitions of the Europarl (EU) and the News-Commentary (NC) corpora (K denotes $\times 1.000$, and M denotes $\times 1.000.000$).

EU+NC	En	Es	En	Fr	En	De
sent.pairs (M)	1.0		1.0		1.0	
average length (words)	21	22	21	23	22	21
vocabulary size (Kwords)	88	121	85	96	82	246
running words (Mwords)	21	22	20	22	23	21
singletons (Kwords)	37	46	35	35	32	124

of the task.

A special comment needs the perplexity figures in Tables 3.4 and 3.5. These figures were obtained with language models trained in their corresponding partitions, being these models also used in the translation process. Perplexity figures are an appealing indicator of the complexity of the development and test sets from the point of view of the language model. For instance, they reflect the complexity of German in the Europarl corpus and the out-of-domain nature of the News-Commentary corpus with respect to the partition on which the language model was trained.

Experimental setting and results

As mentioned before, the M1 mixture model was indirectly evaluated on the translation quality of a phrase-based system generated from the Viterbi alignments of this model. The publicly available Moses toolkit [K$^+$07], which implements the log-linear approach to statistical MT, was employed to train phrase-based systems from Viterbi alignments.

In our experiments, the log-linear combination involved the conventional baseline components integrated into the Moses multi-stack decoder:

- Phrase model (direct and inverse phrase and lexical scores, and phrase penalty).
- Distance-based reordering model.
- Lexicalised reordering model.
- Language model.
- Word penalty.

Apart from the decoder, this toolkit provides a series of powerful scripts, which are abundantly employed in this thesis with the following functionality:

- Training of phrase and lexicalised reordering tables from Viterbi alignments.

- Adjustment of weights of the log-linear model according to minimum error rate training (MERT) criterion.
- Phrase and lexicalised reordering table filtering.
- Automatic evaluation of translation quality using BLEU score.

Phrase and lexicalised reordering tables were trained with Viterbi alignments computed after 5 iterations of the M1 mixture model ($mix\, 1^5$). The average computing time[a] is approximately 15 minutes per M1 iteration and component. As far as the language model is concerned, we trained smoothed word-based 5-gram interpolated models with modified Kneser-Ney discount [CG96] using the SRILM toolkit [Sto02] on the monolingual version of the Europarl corpora for English (En), Spanish (Es), French (Fr) and German (De).

Concerning the weights of the components of the log-linear model, we tuned those weights on the development set according to the MERT criterion for the phrase-based system resulting from the HMM Viterbi alignments (see chapter 5). Then, the same weighting scheme was employed for all the experiments in the same language pair throughout the different translation models (M1, M2 and HMM) and over different number of components ($T = 1, 2, 3$), as well as for the baseline system. The same experimental conditions were used to translate both test development sets, Europarl and News Commentary.

At this point, we would like to make clear that statistical phrase-based systems are not one of the scientific goals of this thesis. Therefore, we will be using Moses as a black box[b] in which first we input the Viterbi alignments for the training set provided by our word alignment translation models (M1, M2 or HMM mixture models), then we tune the weights of the log-linear model (if necessary), and finally we obtained as an output a translation for each sentence in the test set.

BLEU scores are reported in Tables 3.8 and 3.9 as a function of the number of components in the M1 mixture model on the preprocessed development test sets of the Europarl and News Commentary corpora, respectively. The column labelled as *baseline* stands for the baseline system proposed in the shared-task ACL 2007 statistical MT workshop, training the translation model on the concatenation of the Europarl and News-Commentary corpora. The basic difference between the baseline system and our system is the training scheme of the word-based alignment models employed to compute the Viterbi alignments. In the case of the baseline system, the training scheme is $1^5 3^3 4^3$, that is, 5 iterations of the M1 model, 3 iterations of the M3 model and 3 iterations of the M4 model using the GIZA++ toolkit, and mkcls [Och99] to generate word classes needed in the training process. The baseline system provides BLEU reference figures at the level of state-of-the-art in statistical MT to which we can compare the translation quality of our translation models (M1, M2 and HMM mixture models).

[a] On a 2.0 GHz Intel Xeon machine

[b] Default parameters are used, unless it is explicitly stated otherwise.

The results offered by the M1 mixture model are far from those of the baseline system, as we could foresee from the dissimilar training schemes. However the analysis of the evolution of the BLEU score as a function of the number of components in the M1 mixture model is the focus of the study of the figures presented in Tables 3.8 and 3.9. In Table 3.8, we can observe a slight improvement, not statistically significant, when we move from the conventional single-component M1 model to the multiple-component M1 mixture model on the Europarl test. Nevertheless, this is not the case for the News-Commentary test set (see Table 3.9) in which there is no gain when increasing the number of components in the M1 mixture model, except for the English-French direction that shows an increase of half a point in BLEU.

Table 3.8: BLEU scores on the Europarl development test partition for the *baseline* system and the M1 mixture model ($T = 1, 2, 3$).

BLEU	*baseline*	1	2	3
En-Es	31.6	29.1	29.2	29.2
Es-En	32.1	29.9	30.0	30.0
En-Fr	31.1	28.4	28.6	28.6
Fr-En	32.2	29.0	29.1	28.1
En-De	19.1	17.4	17.5	17.5
De-En	26.8	24.4	24.4	24.5

Table 3.9: BLEU scores on the News-Commentary development test partition for the *baseline* system and the M1 mixture model ($T = 1, 2, 3$).

BLEU	*baseline*	1	2	3
En-Es	31.2	24.7	24.7	24.6
Es-En	32.5	27.6	27.6	27.6
En-Fr	24.7	19.4	19.5	19.9
Fr-En	25.2	21.0	20.8	20.8
En-De	14.1	11.4	11.4	11.4
De-En	20.8	17.6	17.4	17.4

3.6 Conclusions and future work

In this chapter, we have reviewed and derived the well-known M1 model, before introducing its mixture version. The M1 mixture model aims at capturing context-specific translation processes that are common in natural languages, but had not been directly addressed so far in the literature. The M1 model presented in this chapter bridges the gap between bilingual TC and statistical MT by being a common ingredient in both applications.

In bilingual TC, the M1 model is revealed as an effective approach to apprehend the word correlation across languages in bilingual documents. Doing so, we outperformed the accuracy of those bilingual classifiers that considers each language separately. To be precise, the unigram-M1 model was statistically significantly superior to the bilingual unigram classifier on the BAF corpus.

Apart from the unigram-M1 mixture model, we experienced with evolutions of this model replacing the M1 model by the M2 model, defining in this way a non-uniform alignment probability distribution. Specifically, the unigram-M2 and the bigram-M2 mixture models were derived, implemented and informally evaluated. Both of them suffer even more severely from data scarcity problems than the unigram-M1 model, and their performance was worse than that of the unigram-M1 model. Nevertheless, they served as inspiration of the M2 mixture model that will be introduced in Chapter 4.

A straightforward extension of the model presented in this chapter is the replacement of the unigram language model by higher order language models with richer context information. We believe that this extension could provide an adequate tradeoff between cross-lingual word correlation and context information with promising results. Furthermore, we plan to incorporate bilingual classes [Och99] in order to control the model complexity in the presence of data spareness by adjusting the number of word classes.

In statistical MT, we revisited the computation of the Viterbi alignments for the M1 model and explained how we extended it for the M1 mixture model. As shown in Chapter 1, the Viterbi alignments are the foundations for statistical phrase-based systems, therefore we exploited this idea in order to assess the alignment and translation quality of the M1 mixture model. The evaluation of alignment quality on the Hansard task did not show a clear contribution of applying mixture modelisation to the M1 model. In the case of the evaluation of the translation quality, we employed the Moses toolkit to generate phrase-based systems from the Viterbi alignments trained on the concatenation of the Europarl and News-Commentary corpora. The results obtained reflect minor, but systematic improvements in BLEU scores on the Europarl development test, that encouraged us to develop the M2 mixture model in Chapter 4.

The work related to the unigram-M1 mixture model for bilingual TC have been submitted to an international conference:

- **J. Civera** and A. Juan. Bilingual Text Classification using the IBM 1 Translation Model. Accepted for publication in the sixth international conference on Language Resources and Evaluation, LREC 2008.

BIBLIOGRAPHY

[AD06] N. F. Ayan and B. J. Dorr. Going beyond AER: an extensive analysis of word alignments and their impact on MT. In *Proc. of CONLING/ACL'06*, pages 9–16, Morristown, NJ, USA, July 2006. Association for Computational Linguistics.

[B$^+$93] P. F. Brown et al. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[CB$^+$07] C. Callison-Burch et al. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[CG96] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proc. of ACL'96*, pages 310–318, Morristown, NJ, USA, June 1996. Association for Computational Linguistics.

[FM07a] A. Fraser and D. Marcu. Getting the structure right for word alignment: LEAF. In *Proc. of EMNLP-CoNLL'07*, pages 51–60, Morristown, NJ, USA, June 2007. Association for Computational Linguistics.

[FM07b] A. Fraser and D. Marcu. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303, 2007.

[K$^+$07] P. Koehn et al. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL'07: Demo and Poster Sessions*, pages 177–180, Morristown, NJ, USA, June 2007. Association for Computational Linguistics.

[KS07] P. Koehn and J. Schroeder. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227, Morristown, NJ, USA, June 2007. Association for Computational Linguistics.

[Och99] F. J. Och. An efficient method for determining bilingual word classes. In *Proc. of EACL'99*, pages 71–76, Morristown, NJ, USA, June 1999. Association for Computational Linguistics.

[ON03] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[Sto02] A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proc. of ICSLP'02*, pages 901–904, September 2002.

[ZX06] B. Zhao and E. P. Xing. BiTAM: Bilingual Topic AdMixture Models for Word Alignment. In *Proc. of COLING/ACL'06*, Morristown, NJ, USA, July 2006. Association for Computational Linguistics.

CHAPTER 4

MIXTURE OF M2 MODELS

4.1 Introduction

In this chapter we describe a mixture extension of the M2 model, second model of the well-known IBM translation models [$B^{+}90$, $B^{+}93$], along with its corresponding EM parameter estimation. The M2 model is a refinement of the M1 model in which the uniform alignment probability distribution is replaced by a probability distribution depending on the source position and the target sentence length[a].

The M2 mixture model was evaluated on two statistical MT tasks. For the first task, a dynamic-programming search algorithm for the M2 mixture model was implemented, as a mixture extension of that presented in [GVCN98, GVC01]. For the second task, as we did for the M1 mixture model, we computed the Viterbi alignments for the M2 mixture model that were employed to train a phrase-based system.

The organisation of this chapter is as follows. Section 4.2 introduces the M2 model, and its mixture extension is studied in Section 4.3. Then, we discuss the dynamic-programming search algorithm in Section 4.4, presenting results on a small task in Section 4.5. Further experimental results on two large scale tasks are reported in Section 4.6. Finally, we conclude in Section 4.7.

4.2 The M2 model

4.2.1 The model

The derivation of the M2 model is almost the same to that of the M1 model in Section 3.2.1. The main difference between M1 and M2 models resides in the assumption that is made to define the alignment probability distribution

$$p(a_j \mid x_1^{j-1}, a_1^{j-1}, y) := p(a_j \mid j, |y|) \tag{4.1}$$

[a] Actually, the alignment probability distribution also depends on the source sentence length, but we have dropped this dependency to simplify the model.

where $p(a_j \mid j, |y|)$ is the alignment probability distribution, replacing the uniform alignment distribution of the M1 model in Eq. (3.3). So, in contrast to Eq. (3.5), we have

$$p(x, a \mid , y; \mathbf{\Theta}) = \prod_{j=1}^{|x|} p(a_j \mid j, |y|)\, p(x_j \mid y_{a_j}) \tag{4.2}$$

where $\mathbf{\Theta}$ is defined to also include the alignment parameters of the M2 model

$$\mathbf{\Theta} = \begin{cases} p(i \mid j, |y|) & \forall\, i \in \{0, 1, \ldots, |y|\},\ j \in \{1, \ldots, |x|\} \text{ and } |y| \\ p(u \mid v) & u \in \mathcal{X}, v \in \mathcal{Y} \end{cases} \tag{4.3}$$

in contrast to the parameter vector of the M1 model in Eq. (3.6).

As we did in Eq. (3.9) for the M1 model, we can express the M2 model in terms of indicator vectors

$$p(x \mid y; \mathbf{\Theta}) = \prod_{j=1}^{|x|} \sum_{\boldsymbol{a}_j} \prod_{i=0}^{|y|} [p(i \mid j, |y|)\, p(x_j \mid y_i)]^{a_{ji}} \tag{4.4}$$

so, the M2 model becomes

$$p(x \mid y; \mathbf{\Theta}) = \prod_{j=1}^{|x|} \sum_{i=0}^{|y|} p(i \mid j, |y|)\, p(x_j \mid y_i) \tag{4.5}$$

that is the conventional form of this model.

4.2.2 Maximum likelihood estimation

The maximum likelihood estimation of the parameters of the M2 model is performed in an analogous way to that of the M1 model in Section 3.2.2 using the EM algorithm. Generally speaking, we just need to substitute the uniform alignment distribution of the M1 model by the alignment distribution of the M2 model.

The E step computes $a_{nji}^{(k)}$ as in Eq. (3.15), incorporating the M2 alignment distribution

$$a_{nji}^{(k)} = \frac{p(i \mid j, |y_n|)^{(k)}\, p(x_{nj} \mid y_{ni})^{(k)}}{\sum_{i'=0}^{|y_n|} p(i' \mid j, |y_n|)^{(k)}\, p(x_{nj} \mid y_{ni'})^{(k)}} . \tag{4.6}$$

In the M step, we obtain the same update equation for the statistical dictionary as in Eq. (3.16) for the M1 model. Moreover, we need an additional equation for the alignment parameters

$$p(i \mid j, |y|)^{(k+1)} = \frac{N(i, j, |y|)}{\sum\limits_{i'=0}^{|y|} N(i', j, |y|)} \qquad \forall i, j \text{ and } |y| \tag{4.7}$$

where

$$N(i, j, |y|) = \sum_{n=1}^{N} \delta(|x_n| \geq j)\, \delta(|y_n| = |y|)\, a_{nji}^{(k)} . \tag{4.8}$$

Intuitively, Eq. (4.7) can be understood as a normalised partial count of how many times position j is aligned to position i for target sentences of length $|y|$.

4.3 Mixture of M2 models

4.3.1 The model

The M2 mixture model is straightforward given the derivation of the M1 mixture model in Section 3.3.1 and the conventional M2 model in Section 4.2.1. Here, we introduce a component-dependent version of the alignment parameter in Eq. (4.1) to replace the uniform alignment distribution of the component-conditional M1 model in Eq (3.21),

$$p(x, a \,|\, y, t; \mathbf{\Theta}_t) = \prod_{j=1}^{|x|} \prod_{i=0}^{|y|} [p(i \,|\, j, |y|, t) \, p(x_j \,|\, y_i, t)]^{a_{ji}} \tag{4.9}$$

where the component-conditional parameter vector $\mathbf{\Theta}_t$ in Eq. (3.22) for the M1 mixture model is substituted by

$$\mathbf{\Theta}_t = \begin{cases} p(i \,|\, j, |y|, t) & \forall\, i \in \{0, 1, \ldots, |y|\},\, j \in \{1, \ldots, |x|\} \text{ and } |y| \\ p(u \,|\, v, t) & u \in \mathcal{X}, v \in \mathcal{Y} \end{cases} \tag{4.10}$$

4.3.2 Maximum likelihood estimation

The estimation of the E and M steps of the EM algorithm for the M2 mixture model and the M1 mixture model in Section 3.3.2 are alike.

The E step computes $z_{nt}^{(k)}$ in a similar fashion to Eq. (1.45), but using the underlying component-conditional M2 model. On the other hand, the term $a_{njit}^{(k)}$ is computed similarly to Eq. (3.27), but incorporating the component-conditional M2 alignment distribution

$$a_{njit}^{(k)} = \frac{p(i \,|\, j, |y_n|, t)^{(k)} \, p(x_{nj} \,|\, y_{ni}, t)^{(k)}}{\sum_{i'=0}^{|y_n|} p(i' \,|\, j, |y_n|, t)^{(k)} \, p(x_{nj} \,|\, y_{ni'}, t)^{(k)}} . \tag{4.11}$$

In the M step, mixture coefficients are updated as shown in Eq. (1.48) and statistical dictionaries, as in Eq. (3.28). Finally, component-conditional alignment parameters are newly estimated using

$$p(i \,|\, j, |y|, t)^{(k+1)} = \frac{N(i, j, |y|, t)}{\sum\limits_{i'=0}^{|y|} N(i', j, |y|, t)} \qquad \forall i, j, |y| \text{ and } t \tag{4.12}$$

where

$$N(i, j, |y|, t) = \sum_{n=1}^{N} \delta(|x_n| \geq j) \, \delta(|y_n| = |y|) \, z_{nt}^{(k)} \, a_{njit}^{(k)} . \tag{4.13}$$

The asymptotic cost of the M2 mixture training process per iteration is the same that that of the M1 mixture model, i.e. $O(N \cdot T \cdot \overline{|x|} \cdot \overline{|y|})$.

4.3.3 Smoothing

As we did for the component-conditional statistical dictionary in Section 3.3.3, the component-conditional alignment distribution is smoothed at two levels. First, we smooth the component-conditional alignment distribution interpolating with a uniform distribution over the target positions

$$\hat{p}(i \mid j, |y|, t) = (1 - \epsilon)\, p(i \mid j, |y|, t) + \epsilon\, \frac{1}{|y|}. \tag{4.14}$$

Secondly, we smooth the component-conditional alignment distribution (specific distribution) interpolating with the conventional alignment distribution (general distribution)

$$\hat{p}(i \mid j, |y|, t) = \left(1 - \frac{\beta}{\beta + p(j, |y|)}\right) p(i \mid j, |y|, t) + \frac{\beta}{\beta + p(j, |y|)}\, p(i \mid j, |y|). \tag{4.15}$$

In this case, the interpolation coefficient depends on the relative frequency of the event, source position j and target sentence length $|y|$, in the training set, and on the interpolation parameter β. The interpretation of the interpolation parameter β is similar to that of the parameter α in Section 3.3.3, that is, the interpolation coefficient is 0.5 when β is equal to the relative frequency of the event, source position j and target sentence length $|y|$. The parameter β is manually tuned to optimise the evaluation metric in question on the development set.

4.4 Decoding algorithm

In this section, we introduce a mixture extension of a dynamic-programming decoding algorithm of that presented in [GVCN98, GVC01] in order to directly evaluate the translation quality of the M2 mixture model.

In statistical MT, the aim of the decoding algorithm is to search for a target sentence $\widehat{y}$ given a source sentence x

$$\begin{aligned} \widehat{y} &= \arg\max_{y} p(y \mid x) \\ &= \arg\max_{y} p(y)\, p(x \mid y). \end{aligned} \tag{4.16}$$

The search for $\widehat{y}$ has been demonstrated to be an NP-hard problem [Kni99, UM06]. However, several search algorithms have been proposed in the literature to solve this ill-posed problem efficiently: A^* [B$^+$90], stack-decoding [WW97, AO$^+$99], integer-programming [G$^+$01] and dynamic-programming [GVC01, TN03].

In [GVCN98, GVC01], a dynamic-programming search algorithm for the M2 model is proposed, along with some heuristics to accelerate the search process. This same algorithm has been extended in this thesis to deal with the M2 mixture model. The idea behind this extension to the mixture case is straightforward by considering an extra-dimension in the search trellis to independently store the translation score for each component in the mixture.

Specifically, the translation model $p(x \mid y)$ in Eq. (4.16) is instantiated as a M2 mixture model and $p(y)$, for the sake of simplicity in the notation, will be assumed to be a bigram language model. Then, the score associated to the hypothesis $y_1^{|y|}$, given the source sentence x and the target-sentence length $|y|$ is

$$\prod_{i=1}^{|y|} p(y_i|y_{i-1}) \sum_{t=1}^{T} p(t) \prod_{j=1}^{|x|} \sum_{i=0}^{|y|} p(i \mid j, |y|, t)\, p(x_j \mid y_i, t). \tag{4.17}$$

The expression in Eq. (4.17) can be reformulated in terms of two recursive functions lm and tm

$$lm(y_{|y|}, |y|) \sum_{t=1}^{T} p(t) \prod_{j=1}^{|x|} tm(y_{|y|}, |y|, j, t). \tag{4.18}$$

The definition of the recursive functions lm and tm for any partial hypothesis y_1^i being $y_i = v$ is

$$lm(v,i) = lm(\widehat{v}(v,i), i-1)\, p(v|\widehat{v}(v,i)) \tag{4.19}$$

$$tm(v,i,j,t) = tm(\widehat{v}(v,i), i-1, j, t)\, p(i \mid j, |y|, t)\, p(x_j \mid v, t) \tag{4.20}$$

for all $v \in \mathcal{Y}$, $j = \{1, \ldots, |x|\}$ and $i = \{1, \ldots, |y|\}$. The function $\widehat{v}(v,i)$ returns the previous *best* word v' given that v is going to appear next in the target sentence at position i,

$$\widehat{v}(v,i)) = \arg\max_{v' \in \mathcal{Y}} \Big[lm(v', i-1)\, p(v|v') \times$$
$$\times \sum_{t=1}^{T} p(t) \prod_{j=1}^{|x|} \big(tm(v', i-1, j, t) + p(i \mid j, |y|, t)\, p(x_j \mid v, t) + ftm(j, i+1, t) \big) \Big]$$

being ftm a function that estimates the cost of translating from position $i+1$ to the end of the target sentence,

$$ftm(j,i,t) = \sum_{k=i}^{|y|} p(k \mid j, |y|, t)\, p(x_j \mid \tilde{y}_k, t) \tag{4.21}$$

where $\tilde{y}_1^{|y|}$ is an estimation of the best translation for x. Doing so, this decoder computes the most probable translation of x using the maximum approximation.

The base case of recursion for functions lm and tm is

$$lm(v,1) = p(v|\$) \tag{4.22}$$

$$tm(v,1,j,t) = p(0 \mid j, |y|, t)\, p(x_j \mid NULL, t) + p(1 \mid j, |y|, t)\, p(x_j \mid v, t) \tag{4.23}$$

for all $v \in \mathcal{Y}$ and $j = 1, \ldots, |x|$ and where \$ represents the starting symbol for the language model.

The estimation of the function ftm poses a problem when the target sentence is unknown. A mixture extension of the initial optimistic estimation of ftm, proposed in [GV03], can be calculated as

$$ftm(j, i, t) = \sum_{k=i}^{|y|} \max_{v \in \mathcal{Y}} p(k|j, |y|, t)\, p(x_j|v, t). \tag{4.24}$$

Once a translation has been computed, the function ftm can be re-estimated using this translation. Therefore, the search process includes an iterative refinement process that updates ftm in each iteration. This iterative translation process runs until convergence (when the function ftm remains the same between two consecutive iterations) or for a fixed number of rounds, whatever comes first.

The asymptotic cost of the decoding algorithm for each round is $O(|y| * \mathcal{Y}^m)$, where m is the order of the smoothed n-gram language model. As can be deduced from this cost, the size of the target vocabulary is a critical factor in the decoding time of the algorithm.

4.4.1 Decoding parameters

The decoding algorithm defined in the previous section presents two main difficulties in order to be able to run experiments in a reasonable period of time, even with simple tasks.

First, the search space explores all the words in the target vocabulary, even if many of these words are improbable translations of the words in the source sentence. In order to reduce the cost of the algorithm, only a set of *promising* target words will be considered during the search process. The size of this set is indirectly defined by means of the number of most probable translations W for each word in the source sentence, and the number of “zero-fertility” words WZ.

The set of W-most probable translations S_w is computed according to the inverse translation probability [AO$^+$99]

$$p(v \,|\, u) = \frac{p(u \,|\, v)\, p(v)}{\sum\limits_{u'} p(u' \,|\, v)\, p(v)}$$

that has been adapted for the case of our mixture model

$$p(v \,|\, u) \approx \frac{\sum\limits_{t=1}^{T} [p(t)\, p(u \,|\, v, t)]\ p(v)}{\sum\limits_{u'} \sum\limits_{t=1}^{T} [p(t)\, p(u' \,|\, v, t)]\ p(v)} \tag{4.25}$$

where $p(v)$ is a unigram language model learnt on the training partition. The computation of S_w for a source sentence x is

$$S_w = \bigcup_{j=1}^{|x|} \operatorname*{arg\,max}_{S\subset\mathcal{Y}:|S|=W} \ \min_{v\in S} p(v \mid x_j) \tag{4.26}$$

where we just take the union of the set of W-most probable inverse translations of each source word in the sentence to be translated. The set of "zero-fertility" words S_{wz} is constituted by the WZ-least aligned target words to any source word in the training set, according to the Viterbi alignment for the M2 mixture model. It is necessary to take into account those target words that rarely occur as direct translation of words in the source sentence, otherwise they would not appear in the translated sentence.

The Viterbi alignment for the M2 mixture model is computed in the same way to the M1 mixture model (see Section 3.5.1)

$$\hat{a} \approx \operatorname*{arg\,max}_{a} \max_{t=1,\ldots,T} \ p(t) \prod_{j=1}^{|x|} \max_{a_j} p(a_j \mid j, |y|, t) p(x_j \mid y_{a_j}, t) \tag{4.27}$$

Thus, the set of WZ-least aligned target words is defined as

$$S_{wz} = \operatorname*{arg\,max}_{S\subset\mathcal{Y}:|S|=WZ} \ \min_{v\in S} \sum_{n=1}^{N} \sum_{i=1}^{|y_n|} \delta(v, y_{ni})\, \phi(\hat{a}_n, i) \tag{4.28}$$

where

$$\phi(a,i) = \begin{cases} 0 & \exists\, j : a_j = i \qquad j = 1,\ldots,|a| \\ 1 & \text{otherwise} \end{cases} \tag{4.29}$$

is the condition that says whether the position i is connected to any source position j or not, and δ is the Kronecker function[b].

Finally, the union of the sets S_w and S_{wz} defines the final bag-of-words of candidate target words.

Secondly, the alignment distribution of M2 model depends on the target sentence length, so the decoding algorithm needs to know *a priori* the length of the target sentence that will be output. In practise, this fact implies the need of exploring a range of *promising* target sentence lengths given the source sentence.

The adopted solution considers a Gaussian distribution over the target sentence length depending on the source sentence length. So, the range goes from $\overline{|y|}_{|x|} - L$ to $\overline{|y|}_{|x|} + L$, where $\overline{|y|}_{|x|}$ is the average length of the target sentence given the length of source sentence to be translated and L is a parameter that controls the range width. This range width is a factor that multiplies the asymptotic cost of the algorithm.

[b] $\delta(a,b)$ is 1 if $a = b$ and zero otherwise.

Another useful parameter to control the response time of the decoding algorithm is the maximum number of search rounds D. This parameter defines the number of times that the same source sentence is going to be translated for a fixed target-sentence length. For each round the function ftm is recomputed.

Finally, a beam-search parameter B is set in order to prune those hypotheses whose score was lower than the best score multiplied by this parameter.

All the parameters presented in this section were tuned in order to control the trade-off between translation quality and response time, preserving the benefits of using more components in the M2 mixture model.

4.5 Experimental results

The Spanish-English TOURIST task [ABC$^+$00] was selected to assess the M2 mixture model. It is composed of sentence pairs corresponding to human-to-human communication situations at the front-desk of a hotel which were semi-automatically produced using a small seed corpus compiled by four persons from travel guides booklets dealing with different topics. A corpus of $10,000$ random sentences pairs was selected for training purposes and a test partition was defined using $2,996$ random sentence pairs generated independently from the training partition. The basic statistics of this corpus are shown in Table 4.1.

Table 4.1: Basic statistics of the Spanish-English TOURIST task (K denotes $\times 1.000$).

	Training Set		Test Set	
	Es	En	Es	En
sentences	10.000		2.996	
average length	9	9	11	11
vocabulary size	686	513	611	468
singletons	10	8	63	49
running words	97K	99K	35K	36K
perplexity	-	-	-	4.92

This multimodal corpus defines an excellent test bed to evaluate the M2 mixture model, since its simplicity will bring about the pros and cons of the model.

Several experiments were carried out with the Spanish-English TOURIST task to analyse the evolution of the error rate as a function of the number of mixture components ($T \in \{1, 2, 5, 10, 20\}$). On the one hand, the training process starts by iterating with the M1 mixture model from a random initialisation until convergence. Then, the parameters learnt in the M1 mixture model are transferred to the M2 mixture model that is also trained until convergence. This two-step procedure favours a smoothed parameter learning from a simpler model to a more complex model. On the other hand, the search parameters were fixed in order to not interfere in the study of the translation model itself, so that a large number of hypotheses were explored. The language models used in these experiments were smoothed bigrams and trigrams based on back-off with Witten-Bell discount [WB91].

Figure 4.1 shows the evolution of the WER[c] (left y axis) and BLEU score (right y axis), on the test partition of the TOURIST task, for an increasing number of mixture components (x axis). Each curve represents the progress of an evaluation measure, WER (W) or BLEU (B), when using a smoothed bigram (2g) or trigram (3g) language model. Each plotted point is an average over values obtained from 10 randomised trials.

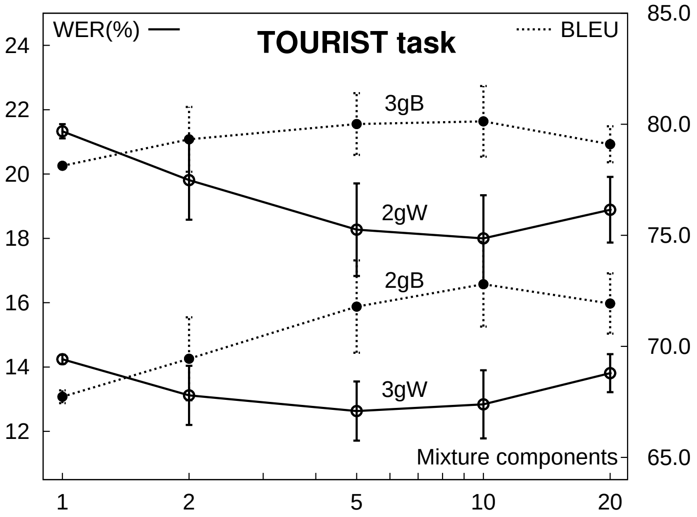


Figure 4.1: WER (W) and BLEU (B) curves in the test partition as a function of the number of mixture components using smoothed bigram (2g) and trigram (3g) language models.

When analysing the results in Figure 4.1, it is clearly observed a systematic WER decrease (BLEU increase) as more components are added to the M2 mixture model. This positive trend reverts at different parameter settings depending on the language model we are using. In the case of bigrams that happens when the model incorporates 20 components into the mixture, while in trigrams this trend reverts when using 10 components. The reason behind this behaviour is mainly due to the fact that a trigram language model leaves less space for improvement than a simple bigram language model. So, the refinement of the translation model through the incorporation of more components produces greater benefit in a simpler bigram model, than in an already sophisticated trigram model.

[c] As a reference, the single-component version of these results are correlated with those obtained in [GV03].

A summary of single-component and best mixture results for bigram and trigram language models is shown in Table 4.2. These figures reflect that the M2 mixture model provides an average relative improvement in WER of 15% for the bigram language model and 11% for the trigram language model, w.r.t. the single-component M2 model. These improvements are statistically significant.

Table 4.2: Baseline and best mixture results on the Spanish-English TOURIST task. The n-column indicates the n-gram order of the language model, while the T-column denotes the number of components in the M2 mixture model.

n	T	WER	BLEU
2	1	21.3	67.7
	10	18.0	72.8
3	1	14.2	78.1
	5	12.6	80.0

We are aware that the results reported in this section are far from those obtained in the same corpus with state-of-the-art phrase-based models, specifically Alignment Templates [Och99], and stochastic finite-state transducers, more precisely GIATI [CV04]. However, these experiments allow us to study the behaviour of the M2 mixture model under controlled experimental conditions, i.e. simple task and customised decoder, avoiding so the influence of other external factors that could mask the results.

4.6 Further evaluation

The evaluation of the M2 mixture model presented in the previous section provides a direct insight into the capabilities and properties of the model. However, as we said before, the results reported are far from those obtained with state-of-the-art phrase-based systems. Furthermore, the conclusions drawn from the results on the small synthetic TOURIST task are difficult to be extrapolated to other corpora. To have a broader view of the alignment and translation quality of the model, we performed a throughout evaluation on the shared tasks presented in Section 3.5.

The corpora and experimental setting of this chapter are identical to that of Section 3.5, except for the training scheme that was used to compute the Viterbi alignments. In this case, the training scheme was $mix\,1^5 2^5$, that is, 5 iterations of the M1 mixture model followed by 5 iterations of the M2 mixture model. As usual, the statistical dictionary learnt by the M1 mixture model are transferred to the M2 mixture model. For the joint Europarl and News Commentary training corpus, the average computing time[d] is approximately 20 minutes per M2 iteration

[d] On a 2.0 GHz Intel Xeon machine

and component. It should be borne in mind that the weights of the log-linear model integrated in Moses are the same to those in Chapter 3.

4.6.1 Evaluation of alignment quality

Table 4.3 presents AER figures on the test partition for the M2 mixture model. As in case of the M1 mixture model, each number in Table 4.3 is an average over values obtained from 10 randomised initialisation. The computation of the Viterbi alignments was calculated according to Eq. (4.27).

Table 4.3: AER figures on the test partition of the Hansard corpus for the M2 mixture model varying the number of components in the mixture ($T = 1,\ 2,\ 3$) and the conventional M2 model implemented in the GIZA++ toolkit.

	GIZA++	1	2	3
Fr-En	20.0	19.6	19.0	18.8
En-Fr	18.3	17.6	17.2	16.8

As seen in Table 4.3, there is a statistically significant improvement when we go from the conventional single-component M2 model to the multiple-component M2 mixture model for both language directions. Furthermore, the decrease in AER on the English-French direction when we increase from two to three the number of components is also statistically significant.

4.6.2 Evaluation of translation quality

BLEU scores are reported in Tables 4.4 and 4.5 as a function of the number of components in the M2 mixture model on the preprocessed development test sets of the Europarl and News Commentary corpora, respectively. As it happened in the M1 mixture model, the BLEU scores reported are far from the baseline system. This is due to the more refined models (M3 and M4) employed in the baseline system compared to the relatively simple M2 model.

Furthermore, in contrast to the appealing results presented in Section 4.5, there is little gain in BLEU score on the Europarl development test set when increasing the number of components per mixture. Nonetheless, there is an average systematic increase of half a point in BLEU score on the News-Commentary development test set (except for the English-German direction), when using the Viterbi alignments provided by 2-component or 3-component M2 mixture models at the backend of Moses. These improvements are not statistically significant.

4.7 Conclusions and future work

In this chapter we presented a mixture extension of the M2 model together with its maximum likelihood parameter estimation and a specific decoding algorithm.

Table 4.4: BLEU scores on the Europarl development test partition

BLEU	*baseline*	1	2	3
En-Es	31.6	30.7	31.1	30.2
Es-En	32.1	31.4	30.9	31.5
En-Fr	31.1	30.4	30.5	30.8
Fr-En	32.2	31.3	31.5	31.5
En-De	19.1	19.0	18.9	19.0
De-En	26.8	26.0	26.1	26.0

Table 4.5: BLEU scores on the News-Commentary development test partition

BLEU	*baseline*	1	2	3
En-Es	31.2	29.2	29.6	29.6
Es-En	32.5	31.2	30.9	31.7
En-Fr	24.7	23.4	23.9	23.9
Fr-En	25.2	23.7	23.9	24.3
En-De	14.1	13.0	12.8	12.9
De-En	20.8	19.3	19.6	19.5

The experiments conducted on a small synthetic task, clearly indicate the benefits of the mixture approach over the single-component M2 model. Even though these results are not competitive enough compared to those obtained by state-of-the-art phrase-based models in the same task. These experiments were complemented with results on alignment and translation shared-tasks, already introduced in Chapter 3, in order to study the behaviour of the model in real tasks. The results on alignment quality showed a statistically significant improvement as we increase the number of components in the mixture model. Regarding the translation quality of the M2 mixture model, the BLEU figures revealed a systematic average increase of half a point in most of the language pairs of the News-Commentary development test, although this improvement was not conveyed to the Europarl development test set.

The BLEU scores reported in this chapter are still behind those of the baseline system. This fact leads us to consider superior word alignment translation models to bridge this gap in performance. However, the complexity of these superior models should be moderate so as to avoid overtraining. Taking this concern into account, we introduce the HMM alignment model in Chapter 5.

The work related to the M2 mixture model using the dynamic-programming search algorithm presented was published in an international conference:

- **J. Civera** and A. Juan. Mixtures of IBM Model 2. In *Proceedings of the 11th* *annual conference of the European Association for Machine Translation, EAMT 2006*, pages 159–167, Oslo (Norway), June 2006.

The AER results on the Hansard task presented in this chapter will be published in an international workshop:

- **J. Civera** and A. Juan. Word alignment quality in the IBM 2 mixture model. In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems, PRIS 2008*, INSTICC Press, Barcelona (Spain), June 2008.

BIBLIOGRAPHY

[ABC$^+$00] J.C. Amengual, J.M. Benedí, F. Casacuberta, A. Castaño, A. Castellanos, V. Jiménez, D. Llorens, A. Marzal, M. Pastor, F. Prat, E. Vidal, and J.M. Vilar. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103, 2000.

[AO$^+$99] Y. Al-Onaizan et al. Statistical Machine Translation: Final Report. Technical report, Johns Hopkins University 1999 Summer Workshop on Language Engineering, Center for Language and Speech Processing, Baltimore, MD, USA, 1999.

[B$^+$90] P. F. Brown et al. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

[B$^+$93] P. F. Brown et al. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[CV04] F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

[G$^+$01] U. Germann et al. Fast decoding and optimal decoding for machine translation. In *Proc. of ACL'01*, pages 228–235, Morristown, NJ, USA, June 2001. Association for Computational Linguistics.

[GV03] I. García-Varea. *Traducción automática estadística: Modelos de traducción basados en máxima entropía y algoritmos de búsqueda*. PhD thesis, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, Valencia, España, Diciembre 2003.

[GVC01] I. García-Varea and F. Casacuberta. Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *Proc. of MT Summit VIII*, pages 115–120, Santiago de Compostela, Spain, 2001.

[GVCN98] I. García-Varea, F. Casacuberta, and H. Ney. An iterative, DP-based search algorithm for statistical machine translation. In *Proc. of ICSLP'98*, pages 1235–1238, Sydney, Australia, October 1998.

[Kni99] K. Knight. Decoding complexity in word-replacement translation models. *Computional Linguistics*, 25(4):607–615, 1999.

[Och99] F. J. Och. An efficient method for determining bilingual word classes. In *Proc. of EACL'99*, pages 71–76, Morristown, NJ, USA, June 1999. Association for Computational Linguistics.

[TN03] C. Tillmann and H. Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, 29(1):97–133, March 2003.

[UM06] R. Udupa and H. K.r Maji. Computational complexity of statistical machine translation. In *Proc. of EACL'06*, April 2006.

[WB91] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37:1085–1094, 1991.

[WW97] Y. Wang and A. Waibel. Decoding algorithm in statistical translation. In *Proc. of ACL'97*, pages 366–372, Morristown, NJ, USA, July 1997. Morgan Kaufmann / Association for Computational Linguistics.

CHAPTER 5

MIXTURE OF HMM ALIGNMENT MODELS

5.1 Introduction

The HMM alignment model was initially proposed in [V$^+$96] and refined in [ON03]. This model possesses appealing properties, like the simplicity of the first-order word alignment distribution, and the efficient and exact computation of the E-step and Viterbi alignment using a dynamic-programming algorithm. These properties have made this model suitable for extensions [TIM02, LR05] and integration into a phrase-based model [DB05] in the past.

In this chapter, we present a mixture extension of the HMM alignment model, as we did in Chapter 3 for the M1 model, and in Chapter 4 for the M2 model. Similarly to Chapters 3 and 4, an indirect evaluation of the translation quality was carried out on the Europarl and News-Commentary corpora by utilising the Viterbi alignments of the HMM model to train a phrase-based system.

The structure of this chapter is as follows. We first present the HMM alignment model in Section 5.2 and its mixture extension in Section 5.3. We report experimental results on alignment quality for the Hansard task, and on translation quality for the Europarl and News-Commentary corpora in Section 5.4. Finally, conclusions and future work are stated in Section 5.5.

5.2 The HMM alignment model

5.2.1 The model

For the HMM model, we derive the conditional probability $p(x \mid y)$, as we did in Eqs. (3.1) and (3.2). However, the assumption that we make for the alignment probability distribution in the HMM model differs from that in the M1 and M2 models, shown in Eqs. (3.3) and (4.1) respectively. The alignment p.f. in the HMM model includes a dependency on the previous alignment, also known as a first-order dependency, while the lexical p.f. remains the same as in Eq. (3.4).

More precisely, the original formulation of the HMM alignment model assumes that the alignment a_j depends on the previous alignment a_{j-1} and the length of the target sentence $|y|$, so the alignment p.f. becomes

$$p(a_j \mid a_1^{j-1}, x_1^{j-1}, y) := p(a_j \mid a_{j-1}|y|). \tag{5.1}$$

It is interesting to observe the evolution of the alignment p.f. from the M1 model in Eq (3.3) represented by a simple uniform distribution, over the M2 model in Eq. (4.1) that considers a zero-order dependency alignment, to the HMM model in Eq. (5.1) modelling a first-order dependency.

The HMM alignment model can be either derived from the M1 model in Eq. (3.5), or the M2 model in Eq. (4.2), substituting their alignment p.f. by that defined in Eq. (5.1). Thus, we have

$$p(x, a \mid y; \mathbf{\Theta}) = \prod_{j=1}^{|x|} p(x_j \mid y_{a_j})\, p(a_j \mid a_{j-1}, |y|) \tag{5.2}$$

where we suppose that $a_0 = 0$ and

$$\mathbf{\Theta} = \begin{cases} p(i \mid i', |y|) & 1 \le i \le |y|,\ 0 \le i' \le |y| \text{ and } \forall\, |y| \\ p(u \mid v) & \forall\, u \in \mathcal{X} \text{ and } v \in \mathcal{Y} \end{cases} \tag{5.3}$$

is the set of unknown parameters comprising the first-order dependency alignment parameters and the conventional statistical dictionary.

As we proceeded in the previous models presented in this thesis, we can express Eq. (5.2) in terms of indicator vectors as

$$p(x, a \mid y; \mathbf{\Theta}) = \prod_{j=1}^{|x|} \prod_{i=1}^{|y|} p(x_j \mid y_i)^{a_{ji}} \prod_{i'=1}^{|y|} p(i \mid i', |y|)^{a_{j-1i'} a_{ji}} \tag{5.4}$$

with $a_{00} = 1$.

5.2.2 Maximum likelihood estimation

As we did in previous chapters, we revert to the EM algorithm to estimate $\mathbf{\Theta}$ according to the maximum likelihood criterion w.r.t. a set of N independent samples $(X, Y) = ((x_1, y_1), \ldots, (x_N, y_N))^t$. The log-likelihood function of $\mathbf{\Theta}$ is

$$L(\mathbf{\Theta}; X, Y) = \sum_{n=1}^{N} \log \sum_{a_n} p(x_n, a_n \mid y_n; \mathbf{\Theta}). \tag{5.5}$$

Taking (X, Y) as the observed data X, and the alignment data A in Eq. (3.13) as the missing data Z, the E step computes the expected value of the logarithm of

$p(X, A \mid Y)$ w.r.t. the posterior $p(A \mid X, Y; \mathbf{\Theta}^{(k)})$, analogously to Eq. (1.32),

$$\begin{aligned} Q(\mathbf{\Theta} \mid \mathbf{\Theta}^{(k)}) = & \sum_{n=1}^{N} \sum_{j=1}^{|x_n|} \sum_{i=1}^{|y_n|} a_{nji}^{(k)} \, \log p(x_{nj} \mid y_{ni}) \\ & + \sum_{i'=1}^{|y_n|} (a_{nj-1i'} \, a_{nji})^{(k)} \, \log p(i \mid i', |y_n|) \end{aligned} \tag{5.6}$$

with

$$a_{nji}^{(k)} = \frac{\alpha_{nji}\beta_{nji}}{\sum\limits_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj\tilde{\imath}}\beta_{nj\tilde{\imath}}} \tag{5.7}$$

$$(a_{nj-1i'} \, a_{nji})^{(k)} = \frac{\alpha_{nj-1i'} \, p(i \mid i', |y_n|)^{(k)} \, p(x_{nj} \mid y_{ni})^{(k)} \, \beta_{nji}}{\sum\limits_{\tilde{\imath}'=1}^{|y_n|} \sum\limits_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj-1\tilde{\imath}'} \, p(\tilde{\imath} \mid \tilde{\imath}', |y_n|)^{(k)} \, p(x_{nj} \mid y_{n\tilde{\imath}})^{(k)} \, \beta_{nj\tilde{\imath}}} \tag{5.8}$$

being $(a_{nj-1i'} \, a_{nji})^{(k)}$, the posterior probability of aligning the source position $j-1$ to the target position i' and the position j to the position i for the nth sample. So, the recursive functions α and β are defined as

$$\alpha_{nji} = \begin{cases} p(i \mid 0, |y_n|)^{(k)} \, p(x_{nj} \mid y_{ni})^{(k)} & j = 1 \\ \sum\limits_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj-1\tilde{\imath}} \, p(i \mid \tilde{\imath}, |y_n|)^{(k)} \, p(x_{nj} \mid y_{ni})^{(k)} & j > 1 \end{cases} \tag{5.9}$$

$$\beta_{nji} = \begin{cases} 1 & j = |x_n| \\ \sum\limits_{\tilde{\imath}=1}^{|y_n|} p(\tilde{\imath} \mid i, |y_n|)^{(k)} \, p(x_{nj+1} \mid y_{n\tilde{\imath}})^{(k)} \beta_{nj+1\tilde{\imath}} & j < |x_n|. \end{cases} \tag{5.10}$$

The M step finds a new estimate of $\mathbf{\Theta}$, $\mathbf{\Theta}^{(k+1)}$, maximising Eq. (5.6), as in Eq. (1.28), resulting in update equations for the alignments parameters,

$$p(i \mid i', |y|)^{(k+1)} = \frac{N(i, i', |y|)}{\sum\limits_{\tilde{\imath}=1}^{|y|} N(\tilde{\imath}, i', |y|)} \qquad \forall i, i' \text{ and } |y| \tag{5.11}$$

where

$$N(i, i', |y|) = \sum_{n=1}^{N} \delta(|y_n| = |y|) \sum_{j=1}^{|x_n|} (a_{nj-1i'} \, a_{nji})^{(k)} \tag{5.12}$$

and for the statistical dictionary in Eq. (3.16). Intuitively, Eq. (5.11) is a normalised partial count of how many times target positions to which are aligned two consecutive source positions are $a_{j-1} = i'$ and $a_j = i$ for target sentences of length $|y|$.

5.3 Mixture of HMM alignment models

5.3.1 The model

In a similar fashion to Sections 3.3.1 and 4.3, let us consider that $p(x \mid y)$ has been generated by a T-component mixture as in Eq. (3.18), in this case a mixture of HMM alignment models. Here, we rewrite $p(x, a \mid y, t)$ in Eq. (3.21) as a component-conditional version of the HMM alignment model in Eq. (5.4). Thus, we have that the component-conditional HMM model is

$$p(x, a \mid y, t; \mathbf{\Theta}_t) = \prod_{j=1}^{|x|} \prod_{i=1}^{|y|} p(x_j \mid y_i, t)^{a_{ji}} \prod_{i'=1}^{|y|} p(i \mid i', |y|, t)^{a_{j-1i'} a_{ji}} \tag{5.13}$$

where $a_{00} = 1$ and the parameter vector $\mathbf{\Theta}_t$ is defined as

$$\mathbf{\Theta}_t = \begin{cases} p(i \mid i', |y|, t) & \forall 1 \leq i \leq |y|,\ 0 \leq i' \leq |y| \text{ and } |y| \\ p(u \mid v, t) & \forall u \in \mathcal{X} \text{ and } v \in \mathcal{Y}. \end{cases} \tag{5.14}$$

being a component-dependent version of Eq. (5.3).

5.3.2 Maximum likelihood estimation

The log-likelihood function of $\mathbf{\Theta}$ w.r.t. N independent samples for the HMM mixture model is the same to that in Eq. (3.23).

Instantiating the EM algorithm for the HMM mixture model, we compute the Q function as in Eq. (1.44),

$$\begin{aligned} Q(\mathbf{\Theta} \mid \mathbf{\Theta}^{(k)}) = \sum_{n=1}^{N} \sum_{t=1}^{T} z_{nt}^{(k)} \log p(t) + \sum_{j=1}^{|x_n|} \sum_{i=1}^{|y_n|} (z_{nt}\, a_{nji})^{(k)} \log p(x_{nj} \mid y_{ni}, t) \\ + \sum_{i'=1}^{|y_n|} (z_{nt}\, a_{nj-1i'}\, a_{nji})^{(k)} \log p(i \mid i', |y_n|, t). \end{aligned} \tag{5.15}$$

that involves the computation of $z_{nt}^{(k)}$, $(z_{nt}\, a_{nji})^{(k)}$ and $(z_{nt}\, a_{nj-1i'}\, a_{nji})^{(k)}$. First, $z_{nt}^{(k)}$ is calculated as in Eq. (1.45) using the underlying component-conditional HMM model. Secondly, $(z_{nt}\, a_{nji})^{(k)}$ is decomposed as in Eq. (3.26) where

$$a_{njit}^{(k)} = \frac{\alpha_{njit} \beta_{njit}}{\sum_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj\tilde{\imath}t} \beta_{nj\tilde{\imath}t}} \tag{5.16}$$

is a component-dependent version of Eq. (5.7). Lastly, the term $(z_{nt}\, a_{nj-1i'} a_{nji})^{(k)}$ is similarly decomposed to Eq. (3.26)

$$\begin{aligned}(z_{nt}\, a_{nj-1i'}\, a_{nji})^{(k)} &= p(z_{nt}=1, a_{nji}=1, a_{nj-1i'}=1 \,|\, x_n, y_n)\\ &= p(z_{nt}=1 \,|\, x_n, y_n)\, p(a_{nji}=1, a_{nj-1i'}=1 \,|\, z_{nt}=1, x_n, y_n)\\ &= z_{nt}^{(k)}\, (a_{nj-1i'}\, a_{nji} \,|\, t)^{(k)} \end{aligned} \tag{5.17}$$

where

$$(a_{nj-1i'}\, a_{nji} \,|\, t)^{(k)} = \frac{\alpha_{nj-1i't}\, p(i \,|\, i', |y_n|, t)^{(k)}\, p(x_{nj} \,|\, y_{ni}, t)^{(k)}\, \beta_{njit}}{\sum\limits_{\tilde{\imath}'=1}^{|y_n|} \sum\limits_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj-1\tilde{\imath}'t}\, p(\tilde{\imath} \,|\, \tilde{\imath}', |y_n|, t)^{(k)}\, p(x_{nj} \,|\, y_{n\tilde{\imath}}, t)^{(k)}\, \beta_{nj\tilde{\imath}t}} \tag{5.18}$$

is a component-dependent version of Eq (5.8). In this case, the α and β functions are component-dependent versions of those in Eqs.(5.9) and (5.10).

The M step finds an update estimate of $\mathbf{\Theta}$, $\mathbf{\Theta}^{(k+1)}$, as a result of maximising Eq. (5.15). The update equation for the mixture coefficients is that of Eq. (1.48). The new estimate for the component-dependent alignment parameters is a component-conditional version of Eq. (5.11)

$$p(i \,|\, i', |y|, t)^{(k+1)} = \frac{N(i, i', |y|, t)}{\sum\limits_{\tilde{\imath}=1}^{|y|} N(\tilde{\imath}, i', |y|, t)} \qquad \forall i, i', |y| \text{ and } t \tag{5.19}$$

where

$$N(i, i', |y|, t) = \sum_{n=1}^{N} \delta(|y_n| = |y|)\, z_{nt}^{(k)} \sum_{j=1}^{|x_n|} (a_{nj-1i'}\, a_{nji} \,|\, t)^{(k)} \tag{5.20}$$

and the component-dependent dictionaries are updated as in Eq. (3.28). The asymptotic cost of the training procedure per iteration is $O(N \cdot T \cdot \overline{|x|} \cdot \overline{|y|}^2)$, where $\overline{|x|}$ and $\overline{|y|}$ are the source and target average lengths, respectively.

5.3.3 Viterbi alignments

As we did in Eq. (3.48) for the M1 model and in Eq. (4.27) for the M2 model, it is possible to compute the Viterbi alignment for a bilingual pair according to the HMM model. To this purpose, we need to use an efficient dynamic-programming algorithm that can be derived from a recursive function $\tilde{\alpha}$

$$\hat{a} = \operatorname*{arg\,max}_{a=a_1 \ldots a_{|x|}} \tilde{\alpha}_{|x| a_{|x|}} \tag{5.21}$$

where

$$\tilde{\alpha}_{ji} = \begin{cases} p(i \mid 0, |y|)\, p(x_j \mid y_i) & j = 1 \\ \max\limits_{\tilde{\imath}=1\ldots|y|} \tilde{\alpha}_{j-1\tilde{\imath}}\, p(i \mid \tilde{\imath}, |y|)\, p(x_j \mid y_i) & j > 1 \end{cases} \tag{5.22}$$

whose complexity is $O(|x| \cdot |y|^2)$. As we did in the M1 and M2 mixture models, we approximate the Viterbi alignment by maximising over the components of the mixture. Therefore, we have that the complexity of the computation of the Viterbi alignment in a T-component HMM mixture model is $O(T \cdot |x| \cdot |y|^2)$.

5.4 Experimental results

According to [V$^+$96], we consider that HMM alignment probabilities $p(i \mid i', |y|)$ depend only on the jump width. Also the treatment of the NULL word is the same to that presented in [ON03], as well as the probability of aligning to the NULL word p_0, that is optimised on held-out data. In addition to these simplifications, we drop the dependency on the previous alignment and the target sentence length

$$p(a_j \mid a_{j-1}, |y|) := p(a_j). \tag{5.23}$$

It should be noticed that the way in which the alignment parameters are defined makes this model be deficient[a]. However, this assumption greatly simplifies the alignment parameters, while still representing the vital HMM jump width information. As we did in previous models, the alignment distribution was interpolated with a uniform distribution for smoothing purposes.

The evaluation of the HMM mixture model was carried out with the same corpora and experimental setting to that of Sections 3.5 and 4.6, except for the training scheme that was used to compute the Viterbi alignments. The training scheme was $mix\, 1^5 H^5$, that is, 5 iterations of the M1 mixture model followed by 5 iterations of the HMM mixture model. As in the M2 mixture model, the statistical dictionaries are transferred from the M1 to the HMM model. The computation of the Viterbi alignments employed in these experiments is described in Section 5.3.3. For the joint Europarl and News Commentary training corpus, the average computing time[b] is approximately 2.5 hours per HMM iteration and component.

5.4.1 Evaluation of alignment quality

Table 5.1 presents AER figures on the test partition for the HMM mixture model. As in Tables 3.3 and 4.3, each number in Table 5.1 is an average over values obtained from 10 randomised initialisation.

As shown in Table 5.1, the HMM model exhibits a minor, not statistically significant, reduction in AER when we train a 2-component HMM mixture model on

[a] In the sense that this model reserves probability mass for target positions outside the target sentence boundaries.

[b] On a 2.0 GHz Intel Xeon machine

Table 5.1: AER figures on the test partition of the Hansard corpus for the M1, M2 and HMM mixture models varying the number of components in the mixture ($T = 1, 2, 3$) and the conventional M1, M2 and HMM models implemented in the GIZA++ toolkit.

	GIZA++	1	2	3
Fr-En	8.9	8.9	8.8	9.0
En-Fr	8.4	9.1	9.3	9.4

the French-English direction. Although AER increases as it does the number of components on the English-French direction. In the case of the HMM model, the parameter p_0 that defines the probability of aligning to the NULL word, is vital in the performance of the HMM model and is necessary to adjust this parameter on held-out data. We believe that this high dependency of the AER on the value of the parameter p_0 interferes with the possible benefits of mixture modelisation.

5.4.2 Evaluation of translation quality

BLEU scores are reported in Tables 5.2 and 5.3 as a function of the number of components in the HMM mixture model on the preprocessed development test sets of the Europarl and News Commentary corpora, respectively. The weights of the log-linear model in Moses are the same that those in Sections 3.5 and 4.6. The first conclusion, which we can draw from the figures in Tables 5.2 and 5.3, is that there is not significant difference between the BLEU scores of the baseline system and those of the HMM-based system. As observed in Table 5.2, if we compare the BLEU scores of the conventional single-component HMM model to those of the HMM mixture model, it seems that there is little or no gain from incorporating more components into the mixture for the Europarl corpus. Nevertheless, Table 5.3 offers a minor, but systematic improvement in BLEU scores when we increase from one to two the number of components per mixture.

5.5 Conclusions and future work

In the same line to previous chapters, we introduced a mixture version of the HMM alignment model. This model was employed to generate context-specific Viterbi alignments that were directly evaluated, and also input into a phrase-based system to be indirectly assessed. Regarding the direct evaluation of the HMM mixture model through AER figures, we obtained a minor, not statistically significant, reduction when we increase the number of components.

The BLEU scores reported by the HMM-based system are at the level of state-of-the-art in this task, while the benefits of mixture modelling are minor but, as in the M1 and M2 mixture model, systematic in some cases. All in all, we are fully

Table 5.2: BLEU scores on the Europarl development test partition

EU	*baseline*	1	2	3
En-Es	31.6	31.5	31.6	31.7
Es-En	32.1	31.8	31.9	31.9
En-Fr	31.1	31.1	31.2	30.9
Fr-En	32.2	32.2	32.2	32.1
En-De	19.1	19.9	19.8	20.0
De-En	26.8	27.0	26.9	26.8

Table 5.3: BLEU scores on the News-Commentary development test partition

NC	*baseline*	1	2	3
En-Es	31.2	30.9	31.1	31.1
Es-En	32.5	32.2	32.4	32.4
En-Fr	24.7	25.0	25.3	24.4
Fr-En	25.2	24.7	24.7	24.6
En-De	14.1	14.1	14.2	14.1
De-En	20.8	20.7	20.8	21.0

aware that indirectly assessing the translation quality of a model through a phrase-based system is arguable because of the different factors involved that could mask the results [AD06].

One of the challenges of training a mixture of translation models, is the linear growth of the number of parameters to be learnt as we increase the number of components. This is a specially delicate issue in the case of the statistical dictionary due to the potential quadratic number of parameters and its sparcity. In this thesis, we have proposed smoothing techniques to alleviate this problem, although other ideas grounded on the incorporation of monolingual and bilingual classes [Och99] would also be interesting to consider.

Nonetheless, in the advent of larger open-domain corpora, the idea behind context-specific translation models seem to be more than appropriate, necessary. We believe that the idea behind mixture modelling is inherent to the nature of large corpora in which multimodal distributions are frequent. Indeed, the convenience of using a weighted combination of models, instead of a single model trained on massive scale data has already been proved in [BPX$^+$07].

The HMM mixture model and some of the results presented in this chapter were published in an international workshop:

- **J. Civera** and A. Juan. Domain adaptation in statistical machine translation with mixture modelling. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 177–180, Association for Computational Linguistics, Prague (Czech Republic), June 2007.

BIBLIOGRAPHY

[AD06] N. F. Ayan and B. J. Dorr. Going beyond AER: an extensive analysis of word alignments and their impact on MT. In *Proc. of CONLING/ACL'06*, pages 9–16, Morristown, NJ, USA, July 2006. Association for Computational Linguistics.

[BPX$^+$07] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In *Proc. of EMNLP-CoNLL'07*, pages 858–867, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[DB05] Y. Deng and W. Byrne. HMM word and phrase alignment for statistical machine translation. In *Proc. of HLT-EMNLP'05*, pages 169–176. Association for Computational Linguistics, October 2005.

[LR05] A. Lopez and P. Resnik. Improved HMM alignment models for languages with scarce resources. In *Proceedings of the ACL'05: Workshop on Building and Using Parallel Texts*, pages 83–86, Morristown, NJ, USA, June 2005. Association for Computational Linguistics.

[Och99] F. J. Och. An efficient method for determining bilingual word classes. In *Proc. of EACL'99*, pages 71–76, Morristown, NJ, USA, June 1999. Association for Computational Linguistics.

[ON03] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[TIM02] K. Toutanova, H. T. Ilhan, and C. D. Manning. Extensions to HMM-based statistical word alignment models. In *Proc. of EMNLP'02*, pages 87–94, Morristown, NJ, USA, July 2002. Association for Computational Linguistics.

[V$^+$96] S. Vogel et al. HMM-based word alignment in statistical translation. In *Proc. of CL'96*, pages 836–841, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

CHAPTER 6

COMPUTER-ASSISTED TRANSLATION BASED ON STOCHASTIC FINITE-STATE TRANSDUCERS

6.1 Introduction

Information technology advances in modern society have led to the need of more efficient methods of translation. It is important to emphasise that current MT systems are not able to produce ready-to-use text. Indeed, MT systems are usually limited to specific semantic domains and the translations provided require human post-editing in order to achieve a correct high-quality translation.

A way of taking advantage of MT systems is to combine them with the knowledge of a human translator constituting the so-called CAT paradigm. CAT offers different approaches in order to benefit from the synergy between humans and MT systems. In this thesis we focus on the interactive and predictive MT approach to CAT[a]. Under this approach the user can amend the translation offered by the MT system, while the system takes into account these corrections to improve its translation. This protocol of interaction is more comfortable for the translator that can work with a greater freedom to make changes at any time while the translation is in progress.

The interactive and predictive MT approach to CAT has two important aspects: the models need to provide adequate completions and they have to do so efficiently under usability constrains. To fulfil these two requirements, stochastic finite-state transducers (SFST) [V$^+$05b] have been selected since they have proved to be able to provide adequate translations [KAO98, A$^+$00, BR95] and there exist efficient

[a] In this thesis, we will refer to the interactive and predictive MT approach to CAT simply as CAT, whenever it is clear in the context and does not lead to confusion. However, we are aware that there are other approaches to CAT, such as those based on post-editing.

parsing algorithms [V$^+$05a] that can be easily adapted in order to provide completions.

The adaptation, integration and implementation of these parsing algorithms in a CAT framework is the original contribution of this thesis, since they were already introduced and studied in previous work [Wag74, JM99] in which the details of the algorithms can be found. The inference algorithm and the preprocessing module were implemented by external software, but they are presented here to provide a complete view of the CAT system to the reader.

The structure of this chapter is as follows. Next section introduces the general setting for finite-state models in statistical MT. In Section 6.3, the search procedure for interactive and predictive translation is explained. Experimental results are presented in Section 6.4. Finally, conclusions and future work are exposed in Section 6.5.

6.2 Machine translation with finite-state transducers

In contrast to the usual language and translation models of the translation rule in Eq. (1.8), SFSTs [Cas00, PC01, CV04] model the joint distribution $p(x, y)$. Thus this rule becomes,

$$\hat{y} = \arg\max_{y} p(x, y). \tag{6.1}$$

SFSTs constitute an important framework in syntactic PR and NLP. The simplicity of finite-state models has given rise to some concerns about their applicability to real tasks. Specifically in the field of language translation, it is often argued that *natural languages* are so complex that these simple models are never able to cope with the required source-target mappings. However, one should take into account that the complexity of the mapping between the source and target domains of a transducer is not always directly related to the complexity of the domains themselves. Instead, a key factor is the degree of *monotonicity* or *sequentiality* between source and target subsequences of these domains [CVP05]. Finite-state transducers have been shown to be adequate to handle complex mappings efficiently [Ber79]. Also, SFSTs have been successfully applied to many translation tasks in the past [A$^+$00, C$^+$04a].

A SFST $\mathcal{T}$ is defined as a tuple $\langle \Sigma, \Delta, Q, q_0, q_f, \delta, p, f \rangle$ where Σ and Δ are finite sets of source and target symbols respectively, Q is a finite set of states, q_0 is the initial state, $q_f \subseteq Q$ is the set of final states, $\delta \subseteq Q \times \Sigma \times \Delta^\star \times Q$ is the set of transitions, p and f are two functions

$$p : Q \times \Sigma \times \Delta^\star \times Q \to [0, 1] \tag{6.2}$$

$$f : Q \to [0, 1] \tag{6.3}$$

being p, the transition probability function and f, the final-state probability function that satisfy,

$$f(q) \quad + \sum_{(x,\overline{y},q')\in\Sigma\times\Delta^{\star}\times Q} p(q,x,\overline{y},q') = 1 \quad \forall q \in Q. \tag{6.4}$$

Given $\phi(x,y)$, a path with $|x|$ transitions associated with the translation pair $(x,y) \in \Sigma^{*} \times \Delta^{*}$ is a sequence of transitions

$$\phi(x,y) = (q_0, x_1, \overline{y}_1, q_1)(q_1, x_2, \overline{y}_2, q_2)\dots(q_{|x|-1}, x_{|x|}, \overline{y}_{|x|}, q_{|x|}), \tag{6.5}$$

such that $x_1 x_2 \dots x_{|x|} = x$ and $\overline{y}_1 \overline{y}_2 \dots \overline{y}_{|x|} = y$. The probability of a path is the product of its transition probabilities, multiplied by the final-state probability of the last state in the path

$$p(\phi(x,y)) = \prod_{j=1}^{|x|} p(q_{j-1}, x_j, \overline{y}_j, q_j) \cdot f(q_{|x|}). \tag{6.6}$$

The probability of a translation pair (x,y) according to $\phi(x,y)$ is then defined as the sum of the probabilities of all the paths associated with (x,y)

$$p(x,y) = \sum_{\phi(x,y)} p(\phi(x,y)). \tag{6.7}$$

Therefore Eq. (6.1) could be rewritten as

$$\hat{y} = \arg\max_{y} \sum_{\phi(x,y)} p(\phi(x,y)). \tag{6.8}$$

It should be noted that the maximisation problem stated in Eqs. (6.1) and (6.8) is NP-hard [CdlH00]. Nevertheless, adequate approximations can be obtained by means of efficient search algorithms, like Viterbi [Vit67] for the best path

$$p(x,y) \approx \max_{\phi(x,y)} p(\phi(x,y)) \tag{6.9}$$

and the recursive enumeration algorithm (REA) [JM99] for the n-best paths.

6.2.1 Learning finite-state transducers

There are different families of techniques to train a SFST from a parallel corpus of source-target sentences [CV07]. One of the techniques that has been adopted in this thesis is the *grammatical inference and alignments for transducer inference* (GIATI) technique. This technique is in the category of *hybrid methods* which use statistical techniques to guide the SFST structure learning and simultaneously train the associated probabilities.

Given a finite sample of string pairs, the inference of SFSTs using the GIATI technique is performed as follows [CV04, CVP05]:

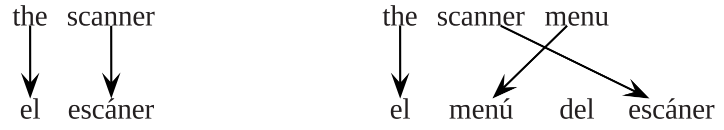
1. Building training strings: each training pair is transformed into a single string from an extended alphabet to obtain a new sample of strings. The "extended alphabet" contains words or substrings from source and target sentences coming from training pairs.

2. Inferring a (stochastic) regular grammar: typically, a smoothed n-gram is inferred from the sample of strings obtained in the previous step.

3. Transforming the inferred regular grammar into a transducer: the symbols associated with the grammar rules are adequately transformed back into source/target symbols.

The transformation of a parallel corpus into a corpus of single sentences is performed with the help of statistical alignments: each word is joined with its translation in the target sentence, creating an "extended word". This joining is done taking care not to invert the order of the target words. The third step is trivial with this arrangement. In our experiments, the alignments are obtained using the GIZA++ software [ON00], which implements the IBM statistical models [B$^+$93].

An example of a SFST is shown in Figure 6.1. This SFST was generated from a bilingual corpus composed of two pairs of sentences manually aligned:



and trained as a smoothed interpolated bigram model on two sentences of extended symbols *the#el scaner#escáner* and *the#el scaner# menu#menú_del_escáner*.

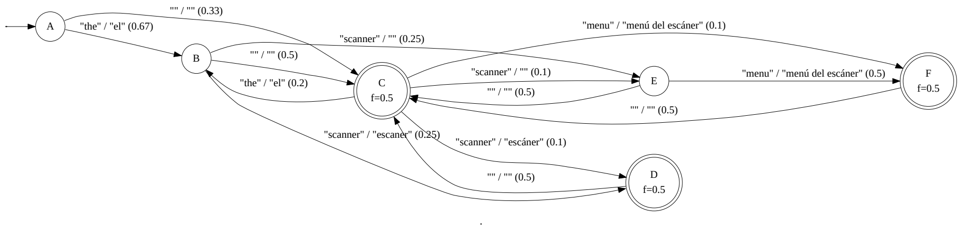


Figure 6.1: Example of the resulting SFST trained on two pairs of sentences: *the scanner # el escáner* and *the scanner menu # el menú del escáner*

6.3 Interactive and predictive search

As commented in Section 6.1, the interactive and predictive approach to CAT proposes a way of interaction based on the target sentence through which the CAT

ITER-0	(y_p)	( )
ITER-1	$(\hat{y}_s)$	(*Haga clic para cerrar el diálogo de impresión*)
	$(\mathbf{a})$	(Haga clic)
	$(\mathbf{k})$	(**en**)
	(y_p)	(Haga clic en)
ITER-2	$(\hat{y}_s)$	(*ACEPTAR para cerrar el diálogo de impresión*)
	$(\mathbf{a})$	(*ACEPTAR para cerrar el*)
	$(\mathbf{k})$	(**cuadro**)
	(y_p)	(Haga clic en ACEPTAR para cerrar el cuadro)
FINAL	$(\hat{y}_s)$	(*de diálogo de impresión*)
	$(\mathbf{a})$	(*de diálogo de impresión*)
	$(\mathbf{k})$	(**#**)
	$(y_p \equiv y)$	(Haga clic en ACEPTAR para cerrar el cuadro de diálogo de impresión)

Figure 6.2: Example of a CAT system interaction to translate into Spanish the English sentence *"Click OK to close the print dialogue"* extracted from a printer manual. Each step starts with a previously fixed target language prefix y_p, from which the system suggests a suffix $\hat{y}_s$. Then the user accepts part of this suffix ($\mathbf{a}$) and types some key strokes ($\mathbf{k}$), in order to amend the remaining part of y_s. This produces a new prefix, composed by the prefix from the previous iteration and the accepted and typed text, $(\mathbf{a})\,(\mathbf{k})$, to be used as y_p in the next step. The process ends when the user enters the special key stroke "**#**". In the final translation, y, all the text that has been typed by the user is underlined.

system and the user exchange portions of it in order to achieve the final correct translation.

An example of this interaction is shown in Figure 6.2. In each iteration, a prefix (y_p) of the target sentence has somehow been fixed by the human translator in the previous iteration and the CAT system computes its best (or n-best) translation suffix hypothesis ($\hat{y}_s$) to complete this prefix.

Given $y_p\hat{y}_s$, the CAT cycle proceeds by letting the user establish a new, longer acceptable prefix. To this end, he or she has to accept a part ($\mathbf{a}$) of $y_p\hat{y}_s$ (or, more typically, just a prefix of $\hat{y}_s$). After this point, the user may type some key strokes ($\mathbf{k}$) in order to amend some remaining incorrect parts. Therefore, the new prefix typically encompasses y_p followed by the accepted part of the system suggestion, $\mathbf{a}$, plus the text, $\mathbf{k}$, entered by the user. Now this prefix, $y_p\mathbf{a}\,\mathbf{k}$, becomes a new y_p, thereby starting a new CAT prediction cycle.

Ergonomics and user preferences dictate exactly when the system can start its new cycle, but typically, it is started after each user-entered word or even after each new user key stroke.

Perhaps the simplest formalisation of the process of hypothesis suggestion of a CAT system is as follows. Given a source text x and a user validated *prefix* of the target sentence y_p, search for a *suffix* of the target sentence that maximises the *posterior* probability over all possible suffixes:

$$\hat{y}_s = \arg\max_{y_s} p(y_s \mid x, y_p)\ . \tag{6.10}$$

Taking into account that $p(y_p, x)$ does not depend on y_s, we can write

$$\hat{y}_s = \arg\max_{y_s} p(y_p y_s, x) \;, \tag{6.11}$$

where $y_p y_s$ is the concatenation of the given prefix y_p and a suffix y_s. Eq. (6.11) is similar to Eq. (6.1), but here the maximisation is carried out over a set of suffixes, rather than full sentences as in Eq. (6.1). Therefore, the model remains the same, that is, we can still use SFSTs, while the search procedure needs to be adequately adapted.

This adapted search procedure has been structured in two phases. The first one copes with the extraction of a word graph $\mathcal{W}$ from a SFST $\mathcal{T}$ given a source sentence x. In a second phase, the search of the best translation suffix (or suffixes) according to the Viterbi approach is performed over the word graph $\mathcal{W}$ given a prefix y_p of the target sentence.

6.3.1 Word graph derivation

A word graph is a compact representation of all the possible translations that a SFST $\mathcal{T}$ can produce from a given source sentence x [C$^+$04c, C$^+$04b]. In fact, the word graph could be seen as a kind of weighted finite-state automaton in which the probabilities are not normalised. Intuitively, the word graph generated retains those transitions in the SFST that were compatible with the source sentence along with their transition probability and output symbol(s).

Formally, given a SFST $\mathcal{T} = \langle \Sigma, \Delta, Q, q_0, q_f, \delta, p, f \rangle$ and a source sentence $x = x_1, \cdots, x_j, \cdots x_{|x|}$, the constructed word graph is defined as a tuple $\mathcal{W} = \langle \Delta, Q', q'_0, q'_f, \delta', p, f \rangle$:

$$\begin{aligned}
Q' &= Q \times j : 0 \leq j \leq |x| \\
\delta' &= \big\{ ((q, j-1), \overline{y}, (q', j)) \mid (q, x_j, \overline{y}, q') \in \delta \big\} \\
q'_0 &= (q_0, 0) \\
q'_f &= \big\{ (q', |x|) \mid ((q, x_{|x|}, \overline{y}, q') \in \delta) \wedge (q' \in q_f) \big\}
\end{aligned}$$

There are several minor issues to deal with in this construction. First, the output symbol for a given transition could contain more than one word. In this case, auxiliary states were created to assign only one word for each transition and simplify the posterior search procedure. Secondly, it is possible to have words in the source sentence that do not belong to the source vocabulary in the SFST. This problem is solved with the introduction of a special generic "unknown word" in the source vocabulary of the SFST. Lastly, if the SFST was generated using a smoothed interpolated language model, then before parsing every word of the source sentence we have to compute all those states reachable with λ-transitions[b] from the set of active states. Also in this case, the set of final states in the word graph is augmented with states that are reachable with λ-transitions from final states.

[b] A λ-transition is a transition of the form $(q, \lambda, \overline{y}, q')$.

An example of a word graph is shown in Figure 6.3. This word graph was extracted from the SFST in Figure 6.1 when parsing the source sentence *the scanner*. The initial step computes those states that are reachable with λ-transitions from the initial state A, that is C. Therefore the set of initial active states S becomes $\{A, C\}$ in the SFST and $W = \{A0, C0\}$ in the word graph. Then, for the first source word *the* and for each state in S, we retain those edges and its associated destiny state whose input symbol is *the*, transferring this information into the word graph. For instance, the edge with input *the* and output *el* going from the state A to the state B in the SFST is mapped into the word graph as the edge with symbol *el* going from the state $A0$ to the state $B1$. As a result of parsing *the*, we define a new set of active states $S = \{B\}$ in the SFST and $W = \{B1\}$ in the word graph, for which we compute those states reachable with λ-transitions, that is, $S = \{B, C\}$ and $W = \{B1, C1\}$. Now we parse the source word *scanner*, transferring those compatible edges and associated states in the SFST to the word graph. For example, the edge going from the state B to the state E with input symbol *scanner* and output symbol λ in the SFST, becomes an edge from the state $B1$ to the state $E2$ with symbol λ. After parsing the last word *scanner*, the set of active states is $S = \{D, E\}$ and $W = \{D2, E2\}$, being these latter states, final states. As we said before, we consider an extra step computing those states reachable with λ-transitions, so we have that $S = \{D, E, C\}$ and $W = \{D2, E2, C2\}$ incorporating the state $C2$ in the set of final states.

Once the word graph is constructed, it can be used to find the best completions for the part of the translation typed by the human translator. Note that the word graph depends only on the source sentence, so it is repeatedly used to find the completions of all the different prefixes provided by the user.

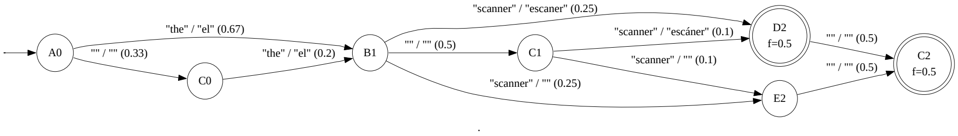


Figure 6.3: Word graph extracted from the SFST in Figure 6.1 when parsing the source sentence *the scanner*.

6.3.2 Search for n-best translations given a prefix of the target sentence

Ideally, the search problem consists in finding the target suffix y_s that maximises the *posterior* probability given a prefix y_p of the target sentence and the source sentence x, as described in Eq. (6.11).

To simplify this search, it will be divided into two steps or phases. The first one would deal with the parsing of y_p over the word graph $\mathcal{W}$. This parsing procedure would end defining a set of states Q'_p that define paths from the initial state whose associated translations include y_p. To clarify this point, it is important to note that each state q in the word graph defines a set of paths reaching this state

$$\varphi_q = \{(q_0, \overline{y}_{q_1}, q_1)(q_1, \overline{y}_{q_2}, q_2), \ldots, (pred(q), \overline{y}_q, q)\} \tag{6.12}$$

and so, a set of translation prefixes

$$\rho(\varphi_q) = \{y'_p = \overline{y}_{q_1}\overline{y}_{q_2} \ldots \overline{y}_q\} \tag{6.13}$$

is defined as the concatenation of the output symbols of the different paths that reach this state q from the initial state. Therefore,

$$Q'_p = \{q : y_p \in \rho(\varphi_q)\}. \tag{6.14}$$

The second phase would be the search of the most probable translation suffix from any of the states in Q'_p. Formally,

$$\psi_q = \{(q, \overline{y}_q, succ(q)), \ldots, (pred(q'), \overline{y}_{q'}, q') : q \in Q'_p \wedge q' \in q'_f\} \tag{6.15}$$

where $y_s = \overline{y}_q \ldots \overline{y}_{q'}$ so

$$\hat{y}_s = \arg\max_{y_s} p(\psi_q). \tag{6.16}$$

Finally, the complete search procedure extracts a translation from the word graph whose prefix is y_p and its remaining suffix is the resulting translation suffix y_s.

Error-correcting parsing

In practise, however, it may happen that y_p is not exactly present in the word graph $\mathcal{W}$. The solution is not to use y_p but a prefix y'_p that is the *most similar* to y_p in some string distance metric. The metric that will be employed is the well-known edit distance based on three basic operations: insertion, substitution and deletion. Therefore, the first phase introduced in the previous paragraph needs to be redefined in terms of the search for those states in $\mathcal{W}$ whose set $\rho(\varphi_q)$ contains y'_p, that is, the set of states Q'_p. It should be noticed that y'_p is not unique, but there exist a set of prefixes in $\mathcal{W}$ whose edit distance to y_p is the same and minimum.

Given a translation prefix y_p, the computation of Q'_p is efficiently carried out by applying an adapted version of the error-correcting algorithm for regular grammars [Wag74] over the word graph $\mathcal{W}$. This algorithm returns the edit distance (ed) with respect to y_p for each state q in $\mathcal{W}$

$$\xi(y_p, q) = \min_{y'_p \in \rho(\varphi_q)} ed(y_p, y'_p). \tag{6.17}$$

However we are interested in those states minimising the edit distance with respect to y_p

$$Q'_p = \arg\min_{q \in Q'} \xi(y_p, q). \tag{6.18}$$

The asymptotic cost of this algorithm is $O(|y_p| \cdot |Q'| \cdot B)$, where B is the (average) branching factor of the word graph $\mathcal{W}$.

The implementation of the error-correcting parsing is further improved by visiting the states in $\mathcal{W}$ in topological order, and incorporating beam-search techniques [Low76] to discard those states whose minimum edit cost is worse than the best minimum edit cost at the current stage of the parsing by a given constant. Moreover, given the incremental nature of y_p, the error-correcting algorithm takes advantage of this peculiarity to parse only the new suffix of y_p provided by the user in the last interaction, that is, the concatenation of $\mathbf{a}$ and $\mathbf{k}$.

As mentioned before, once the set Q'_p has been computed, the search of the most probable translation suffix could be calculated from any of the states in Q'_p. In practise, a subset of states q_p from Q'_p is selected to find the suffix y_s. These states q_p maximise the most the *posterior* probability of the word-graph prefix y'_p computer during the error-correcting parsing process. This maximisation is performed according to the Viterbi approximation.

Furthermore, it may be the case that a user prefix ends in an incomplete word during the interactive translation process. Therefore, it is necessary to start the translation completion with a word whose prefix matches this unfinished word. Thus, the proposed algorithm searches for such a word: first, consider the target words of the edges leaving the nodes returned by the error-correcting algorithm. If this initial search fails, then a matching word is looked up in the word graph vocabulary. Finally, as a last resort, the whole transducer vocabulary is taken into consideration to find a matching word, otherwise this incomplete word is treated as an entire word.

N-best search

The implementation of this CAT system is able to provide a set of different translation suffixes, instead of a single suggestion. To this purpose, an algorithm that searches for the n-best translation suffixes in a word graph is required. Among the n-best algorithms available, the REA described in [JM99] was selected. The main two reasons that support this decision are its simplicity to calculate best paths on demand and its smooth integration with the error-correcting parsing algorithm. Basically, the interaction between these two algorithms, error-correcting and n-best, is carried out by means of the state with the minimum edit distance returned by the error-correcting parsing, from which the n-best translation suffixes can be calculated.

The version of REA included in the CAT system, which is being described, stores for each state q in $\mathcal{W}$, the heap of current best paths (in the form of next state in the best path) from q to any final state. The size of this heap depends on the number of transitions leaving q. During the initialisation of REA, the initial sorted list of best paths for each state is calculated starting from the final states and visiting the rest of states in backward topological order. This last condition imposes a total order in Q' that favours the efficient calculation of the heap of best paths. This is so because each state is visited only once, and once the best paths of the preceding states have already been computed.

Then, among the set of states in Q'_p from which the n-best translation suffixes need to be calculated, REA first extracts the 1-best path from the set of states Q'_p, since it was precomputed during REA initialisation. If $n > 1$, then the next best path will be obtained. The next best path can be found among the candidate paths still left in the heap of states in Q'_p, and the second best path computed from the state from which we extracted the best path.

The computation of the second best path, whenever exists, requires the recursive calculation of this path through the states visited in the 1-best path. This same rationale is applied to the calculation of subsequent best paths until n-best different translation suffixes have been obtained or no more best paths can be found.

6.4 Experimental framework and results

The CAT system introduced in previous sections was assessed through some series of experiments with two different corpora that were acquired and preprocessed in the framework of the TransType2 (TT2) project [Ato01]. In this section, these corpora, the assessment metrics and the results are presented.

6.4.1 XRCE and EU corpora

Two bilingual corpora from different semantic domains were used in the evaluation of the CAT system described. The language pairs involved in the assessment were English/Spanish, English/French and English/German.

The first corpus, namely *XRCE* corpus, was obtained from a miscellaneous set of printer user manuals. Some statistics of the raw version of the corpus are shown in Table 6.1. It should be noted that the English manuals are different in each pair of languages.

The size of the vocabulary in the training set ranges from 25,000 to 37,000 words. In the test set, even though all test sets have similar size, perplexity varies abruptly over the different language pairs.

The second dataset was compiled from the Bulletin of the European Union, which exists in the 11 official languages of the European Union and is publicly available on the Internet. This dataset is known as the *EU* corpus. A summary of its features is presented in Table 6.1.

The size of the vocabulary of EU corpus is at least three times larger than that of the XRCE corpus. These figures together with the amount of running words and sentences reflect the challenging nature of this task. However, the perplexity of the EU test set is lower than that of the XRCE. There are two reasons that combines to explain this phenomenon. First, the different nature of the XRCE and EU corpora, user manuals that required heavy preprocessing versus informative bulletins rather grammatically uniform requiring little preprocessing. Second, the size of the EU corpus is larger than that of the XRCE corpus.

Table 6.1: The XRCE and EU corpora English(En) to/from Spanish(Es), German(De) and French(Fr). Trigrams models were used to compute the test perplexity (K denotes $\times 1.000$, and M denotes $\times 1.000.000$).

		XRCE			EU		
		En/Es	En/Fr	En/De	En/Es	En/Fr	En/De
Train	sent.pairs (K)	56	49	53	214	223	215
	avg.length (words)	10/12	10/11	10/9	24/27	25/28	26/24
	vocabulary (Kwords)	26/30	25/27	25/37	84/97	84/91	86/153
	singletons (Kwords)	10/12	9/11	10/18	38/43	38/40	39/75
	run.words (Mwords)	0.5/0.7	0.5/0.6	0.5/0.4	5/6	5/6	6/5
Test	sentences (K)	1.1	1.0	1.0	0.8	0.8	0.8
	avg.length (words)	7/8	10/10	11/10	25/28	25/28	25/24
	run.words (Kwords)	8/9	10/10	11/10	20/23	20/22	20/19
	run.chars (Kchars)	46/58	55/63	61/71	117/133	117/132	117/132
	perplexity	103/61	180/131	90/155	58/46	58/45	57/87

Corpora preprocessing

A preprocessing module was implemented in order to reduce the corpora complexity and ease the learning process of the models.

The preprocessing has three main parts: tokenisation, lower case conversion and categorisation. Tokenisation basically consisted in the separation of the punctuation marks from the words. After that, all the characters were lowercased. Finally, the categorisation of some types of words was carried out. The idea was to replace those words that remain invariable in all the languages with a category label.

Doing so, the vocabulary size was cut down considerably (up to a 70%), implying an increment in the number of running words (less than 20%). As a result, perplexity decreased significantly, which finally allowed a better transducer inference.

This preprocessing was also applied to the translation process, since the models were learnt on the preprocessed version of the corpora. Consequently, prefixes written by the user had also to be preprocessed. In addition, a postprocess module was needed to make the translations given by the system legible by the user, undoing all the changes introduced by the preprocessing (i.e. uncategorisating, suitably uppercasing and joining punctuation marks to words). Three examples of raw sentences along with their corresponding preprocessed versions extracted from the English partition of the XRCE corpus are shown in Figure 6.4.

1. Chapter 4 Scanning A Document 4-1
chapter <NUM> scanning a document <OTHERS>

2. PRINTS PER CARTON - 2 PACK 6R849
prints per carton <OTHERS> <NUM> pack <OTHERS>

3. 65PPM/230V Sold 109R340
<BULLET> ppm <MIDDLE_SLASH> <OTHERS> sold <OTHERS>

Figure 6.4: Three examples of preprocessed sentences extracted from the English partition of the XRCE corpus. The first line of each pair of sentences shows the raw (original) version of the sentence and the second line, the preprocessed version. As observed some of the tokens that remain invariable were replaced by categorised labels, requiring sometimes a previous tokenisation process. Besides, all characters were converted into lowercase.

6.4.2 Translation quality evaluation

The CAT system was assessed according to two different criteria and therefore, two different sets of evaluation measures are employed.

On the one hand, we proceeded to gauge the translation quality provided by SFST models that lie at the core of the CAT system. These are the so-called *off-line* metrics. This evaluation was performed using WER and BLEU, as in the case of pure statistical translation systems. Here we also computed the *character error rate* (CER) measure, defined as the edit distance in terms of characters between the target sentence provided by the system and the reference translation. CER can be thought of the estimated effort of a fictitious user working with a *dummy* post-editing translation tool that suggests a single translation. This translation would have to be corrected by this fictitious user applying the minimum number of editing operations at the character level to achieve the reference translation.

On the other hand, other assessment figures, namely *on-line* metrics, are aimed at estimating the effort needed by a human translator to produce correct translations using the interactive system. To this end, the target translations which a real user would have in mind are simulated by the given references. The first translation hypothesis for each given source sentence is compared with a single reference translation and the longest common character prefix (LCP) is obtained. The first non-matching character is replaced by the corresponding reference character and then, a new system hypothesis is produced. This process is iterated until a full match with the reference is obtained.

Each computation of the LCP would correspond to the user looking for the next error and *moving the pointer* to the corresponding position of the translation hypothesis. Each character replacement would correspond to a *key stroke* of the user. If the first non-matching character is the first character of the new system hypothesis in a given iteration, no LCP computation is needed; that is, no pointer movement would be made by the user. Bearing this in mind, we define the following interactive-predictive performance measures:

- *Key-Stroke Ratio* (KSR). Number of *key strokes* that should be needed to obtain the reference translation divided by the number of running characters.
- *Mouse-Action Ratio* (MAR). Number of *mouse movements* plus an extra *mouse action* accounting for the acceptance of the final correct translation. A mouse movement is assumed to happen between key strokes which are in non-consecutive positions. It models the effort to position the cursor each time the user would need to amend a part of the system translation.
- *Key-Stroke and Mouse-Action Ratio* (KMSR). KSR plus MAR.

KSR reflects the ratio between the number of key-stroke interactions of a fictitious user when translating a given text using a CAT system compared to the number of key-stroke interactions, which this user would need, to translate the same text without any aiding translation tool. In contrast, the difference between CER and KSR gives us an idea of how much typing effort is saved with the use of an interactive and predictive MT system with respect to a dummy post-editing tool.

The second measure under consideration is KMSR (the calculation of MAR is straightforward given KSR and KMSR) offers a better approximation to the total amount of work that a translator would be saving when translating using a CAT system. In any case, we should keep in mind that the main goal of automatic assessment is to estimate the effort of the human translator. The important question is whether the estimated productivity of the human translator can be increased or not by the CAT approach.

6.4.3 Experimental results

These experimental results were obtained with GIATI transducers based on smoothed trigram language models for the *XRCE* corpus and smoothed 5-gram language models for the *EU* corpus (see Tables 6.2 and 6.3). The translation evaluation measures presented in the previous section were calculated on an independent test set when translating from English into a non-English language and vice versa.

Analysing the results achieved in the *XRCE* corpus (see Table 6.2), it is observed that the results for English-Spanish are substantially better than those obtained in the rest of language pairs. A possible reason that explains these error rate discrepancies between English-Spanish with respect to English-German and English-French could be found in the test perplexity differences shown in Table 6.1. The Spanish test perplexity is significantly lower than that of the rest of languages and this fact is transformed into better translation results.

This rationale is compatible with the results obtained for the *EU* corpus. In these results, the English-Spanish experiment exhibits similar error rates to those of the English-French pairs, but somewhat better than those of the English/German pairs. This same tendency is followed by the perplexity values appearing in Table 6.1. As observed, the German language seems to be more complex than the other languages and this is reflected in Table 6.3.

Table 6.2: Off-line (BLEU, WER[%] and CER[%]) and on-line (KSR, KSMR) measures on the *XRCE* corpora.

	off-line			**on-line**	
XRCE	BLEU	WER	CER	KSR	KSMR
En-Es	52.0	37.9	27.9	13.0	21.8
Es-En	38.9	45.3	32.2	15.9	26.9
En-Fr	24.6	70.7	57.5	30.2	43.8
Fr-En	19.2	68.9	56.1	29.5	45.5
En-De	20.2	74.5	62.9	30.6	45.7
De-En	20.5	71.4	60.6	30.6	46.6

The KSR and KSMR figures of Tables 6.2 and 6.3 clearly manifest a productivity gain if we use the CAT system presented. For example, translating from English into Spanish on the *XRCE* corpus, the user would only need to perform 13.0% of the key-stroke interactions that would be required without this CAT system. On the other hand, the KSR results for the English-French and English-German experiments are 30.2% and 30.6%, respectively. Even in these cases, the number of key-stroke interactions is one third of that that would entail translating the same test set without a CAT system. The results obtained in the other direction are similar.

If we consider the mouse interaction in the CAT evaluation, we can observe a 50% increment in the interaction rates, key strokes plus mouse actions, for most of the language pairs in both corpora. These figures reflect the fact that the productivy gain that CAT systems would theoretically provide is somewhat dependent on the interaction scheme that is assumed.

In the *EU* corpus, the best KSR results were obtained for the English-French experiment, followed by the English-Spanish results and, finally, the worst results were achieved for English-German. Despite the important difference in size between *XRCE* and *EU*, the results are similar and for some language pairs even lower in the *EU* corpus. As previously mentioned, the perplexity figures of both corpora partially explain these results. For instance, the English-French and English-German experiments present lower perplexity figures and better results in the *EU* corpus than in the *XRCE* corpus.

As observed in the result tables, CER figures usually double the KSR figures bringing to light the benefits of an interactive and predictive CAT system compared to a dummy post-editing tool. However, it could be argued that this comparison is not completely fair since the CER measure simulates a very simple post-editing system. We are aware that a real post-editing tool would incorporate additional

Table 6.3: Off-line (BLEU, WER[%] and CER[%]) and on-line (KSR, KSMR) measures on the *EU* corpora.

	off-line			**on-line**	
EU	BLEU	WER	CER	KSR	KSMR
En-Es	39.6	54.6	45.3	21.3	33.0
Es-En	39.8	52.0	43.1	20.0	31.1
En-Fr	41.6	52.8	41.5	19.5	30.1
Fr-En	43.3	47.8	39.2	17.8	28.0
En-De	29.4	64.4	54.6	23.4	35.9
De-En	28.7	66.4	57.7	25.8	39.1

functionalities to reduce the typing effort of the user, for instance the prediction of alternatives to the word that is being corrected. Being so as it is, this comparative statement should be carefully considered taking into account the working conditions that were assumed.

Table 6.4: Comparative table 1-best to 5-best for KSR and KSMR [%] on the *XRCE* corpora.

	1-best		**5-best**	
XRCE	KSR	KSMR	KSR	KSMR
En-Es	13.0	21.8	11.2	19.2
Es-En	15.9	26.9	13.6	23.5
En-Fr	30.2	43.8	27.3	40.1
Fr-En	29.5	45.5	26.9	42.0
En-De	30.6	45.7	27.4	41.8
De-En	30.6	46.6	27.4	42.6

Table 6.4 shows a comparative table between two CAT systems, one of them providing the best translation and the other, 5-best translations. In the latter system, the calculation of KSR and KSMR was conducted considering that translation out of the five suggested translations that minimises the most the number of key strokes needed to achieve the reference translation. As expected, there is a notable improvement when comparing 1-best to 5-best translation accuracy. This gain in translation quality diminishes in a log-wise fashion as we increase the number of best translations.

From a practical point of view, the improvements provided by using n-best translations would come at the cost of the user having to ponder which of these translations is more suitable. In real operation, this additional user effort may or may not outweight the benefits of the n-best increased accuracy. Consequently, this feature should be offered to the users as an option.

In the TT2 project, this CAT system based on SFST was exhaustively evaluated by human translators through real test translation rounds [MNS05, Mac06]. The results showed that the actual productivity of human translators depended on the given test texts. In cases where these texts were quite uncorrelated with the training data, the system did not significantly help the human translators to boost their productivity. However, when the test texts were reasonably well correlated with the training data, a high productivity gain was registered, close to what could be expected according to the KSR/MAR empirical results.

6.5 Conclusions and future work

In this chapter, SFSTs have been revisited and applied to CAT. SFSTs are learnt from parallel corpora and in our case, they were inferred by the GIATI technique, which was briefly reviewed.

Furthermore, the concept of interactive search was introduced along with well-known algorithms, i.e. error-correcting and n-best parsing, that allow us the calculation of the suffix translation that better completes the prefix previously refined by the user. It is fundamental to remember that usability and low response time are vital features for CAT systems. CAT systems need to provide translation suffixes after each user interaction and this imposes the requirement of very efficient algorithms to solve the search problem.

The automatic evaluation carried on two different corpora supports the idea that the incorporation of statistical MT techniques into a CAT system would reduce the human translator effort, without sacrificing the high quality of the translations. This thesis was corroborated by an external evaluation conducted by human translators in real working conditions.

Given the relatively high positioning effort (MAR) observed in the experiments, it seems worth investigating interaction modalities which are alternative or complementary to the traditional keyboard and mouse. In this respect, the use of speech interaction has been considered in [VCR$^+$06], with encouraging results. Finally, the integration of confidence measures [UN05] to guide users' attention into the interactive and predictive CAT scenario are topics still to be explored in future research.

Preliminary versions of the CAT system presented in this chapter has been published in numerous international and national conferences:

- **J. Civera**, J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, F. Casacuberta, E. Vidal, D. Picó and J. González. A syntactic pattern recognition approach to computer assisted translation. In A. Fred, T. Caelli, A. Campilho, R. P.W. Duin, and D. de Ridder, editors, *Advances in Statistical, Structural and Syntactical Pattern Recognition,* Lecture Notes in Computer Science, pages 207–215, Springer-Verlag, Lisbon (Portugal), August 2004.

- **J. Civera**, E. Cubel, A. L. Lagarda, D. Picó, J. González, F. Casacuberta,

E. Vidal, J. M. Vilar and S. Barrachina. From machine translation to computer assisted translation using finite-state models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2004*, pages 349–356, Association for Computational Linguistics, Barcelona (Spain), July 2004.

- E. Cubel, **J. Civera**, J. M. Vilar, A. L. Lagarda, F. Casacuberta, E. Vidal, D. Picó, J. González and L. Rodríguez. From machine translation to computer assisted translation using finite-state models. In *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI 2004*, pages 586–590, IOS Press, Valencia (Spain), August 2004.

- **J. Civera**, E. Cubel, A. L. Lagarda, F. Casacuberta, E. Vidal, J. M. Vilar and S. Barrachina. Computer-assisted translation using finite-state transducers. In *Actas del XXI Congreso de la Sociedad Espaola para el Procesamiento del Lenguaje Natural, SEPLN 2005*, pages 357–363, Granada (Spain), September 2005.

- **J. Civera**, J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, F. Casacuberta and E. Vidal. A novel approach to computer assisted translation based on finite-state transducers. In A. Yli-Jyra, L. Karttunen, and J. Karhumaki, editors, *Finite-State Methods and Natural Language Processing*, Lecture Notes in Artificial Intelligence (LNAI-LNCS). Springer-Verlag, Helsinki (Finland), September 2005.

- **J. Civera**, J. M. Vilar, A. L. Lagarda, E. Cubel, S. Barrachina, F. Casacuberta and E. Vidal. A Computer-Assisted Translation Tool based on Finite-State Technology. In *Proceedings of the 11th annual conference of the European Association for Machine Translation, EAMT 2006*, pages 33–40, Oslo (Norway), June 2006.

However, the content of this chapter reflects the final version of the interactive and predictive CAT system based on SFST technology developed in the TT2 project. This system along with two other CAT systems based on state-of-the-art phrase-based and alignment templates technology and comparative results are to be published in an international journal:

- S. Barrachina, O. Bender, F. Casacuberta, **J. Civera**, E. Cubel, S. Khadivi, A. L. Lagarda, H. Ney, J. Tomás, E. Vidal and J. M. Vilar. Statistical approaches to computer-assisted translation. *Computational Linguistics*, In press.

BIBLIOGRAPHY

[A$^+$00] J. C. Amengual et al. The EuTrans-I speech translation system. *Machine Translation*, 15:75–103, 2000.

[Ato01] Atos Origin, Instituto Tecnológico de Informática, RWTH Aachen, RALI Laboratory, Celer Soluciones and Société Gamma and Xerox Research Centre Europe. TransType2 - Computer Assisted Translation. Project Technical Annex., 2001.

[B$^+$93] P. F. Brown et al. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, 1993.

[Ber79] J. Berstel. *Transductions and context-free languages*. B. G. Teubner Stuttgart, 1979.

[BR95] S. Bangalore and G. Riccardi. A finite-state approach to machine translation. In *Proc. of NAACL'01*, pages 1–8, Morristown, NJ, USA, June 1995. Association for Computational Linguistics.

[C$^+$04a] F. Casacuberta et al. Some approaches to statistical and finite-state speech-to-speech translation. *Computer Speech and Language*, 18:25–47, 2004.

[C$^+$04b] J. Civera et al. From machine translation to computer assisted translation using finite-state models. In Dekang Lin and Dekai Wu, editors, *Proc. of EMNLP 2004*, pages 349–356, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[C$^+$04c] J. Civera et al. A syntactic pattern recognition approach to computer assisted translation. In A. Fred, T. Caelli, A. Campilho, R. P.W. Duin, and D. de Ridder, editors, *Advances in Statistical, Structural and Syntactical Pattern Recognition*, Lecture Notes in Computer Science, pages 207–215. Springer-Verlag, 2004.

[Cas00] F. Casacuberta. Inference of finite-state transducers by using regular grammars and morphisms. In A.L. Oliveira, editor, *Grammatical Inference: Algorithms and Applications*, volume 1891 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag, 2000. 5th International Colloquium Grammatical Inference -ICGI2000-. Lisboa. Portugal.

[CdlH00] F. Casacuberta and C. de la Higuera. Computational complexity of problems on probabilistic grammars and transducers. In A.L. Oliveira, editor, *Grammatical Inference: Algorithms and Applications*, volume 1891 of *Lecture Notes in Computer Science*, pages 15–24. Springer-Verlag, 2000.

[CV04] F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

[CV07] F. Casacuberta and E. Vidal. Learning finite-state models for machine translation. *Machine Learning*, 66(1):69–91, 2007.

[CVP05] F. Casacuberta, E. Vidal, and D. Picó. Inference of finite-state transducers from regular languages. *Pattern Recognition*, 38:1431–1443, 2005.

[JM99] Víctor M. Jiménez and Andrés Marzal. Computing the k shortest paths: a new algorithm and an experimental comparison. In J. S. Vitter and C. D. Zaroliagis, editors, *Algorithm Engineering*, volume 1668 of *Lecture Notes in Computer Science*, pages 15–29. Springer-Verlag, July 1999.

[KAO98] K. Knight and Y. Al-Onaizan. Translation with finite-state devices. In E. Hovy D. Farwell, L. Gerber, editor, *Proc. of AMTA'98*, volume 1529, pages 421–437, London, UK, October 1998. Springer-Verlag.

[Low76] Bruce T. Lowerre. *The harpy speech recognition system*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1976.

[Mac06] E. Macklovitch. TransType2: The last word. In *Proceedings of the 5th International Conference on Languages Resources and Evaluation (LREC 2006)*, pages 167–172, Genoa, Italy, May 2006.

[MNS05] E. Macklovitch, N.T. Nguyen, and R. Silva. User evaluation report. Technical report, TransType2 (IST-2001-32091), 2005.

[ON00] Franz J. Och and Hermann Ney. Improved statistical alignment models. In *Proc. of ACL'00*, pages 440–447, Morristown, NJ, USA, October 2000. Association for Computational Linguistics.

[PC01] David Picó and Francisco Casacuberta. Some statistical-estimation methods for stochastic finite-state transducers. *Machine Learning*, 44:121–142, July-August 2001.

[UN05] N. Ueffing and H. Ney. Application of Word-Level Confidence Measures in Interactive Statistical Machine Translation. In *Proc. of EAMT'05*, pages 262–270, Budapest, Hungary, May 2005.

[V$^+$05a] E. Vidal et al. Probabilistic finite-state machines-part i. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 27(7):1013–1025, 2005.

[V$^+$05b] E. Vidal et al. Probabilistic finite-state machines-part ii. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 27(7):1026–1039, 2005.

[VCR$^+$06] E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. Computer-assisted translation using speech recognition. *IEEE Transaction on Audio, Speech and Language Processing*, 14(3):941–951, 2006.

[Vit67] Andrew Viterbi. Error bounds for convolutional codes and a asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

[Wag74] Robert A. Wagner. Order-n correction for regular languages. *Communications of the ACM*, 17(5):265–268, 1974.

CHAPTER 7
CONCLUSIONS

7.1 Summary

The work developed in this thesis covers three different topics in natural language processing: text classification, statistical machine translation and computer assisted translation.

In TC, we proposed a novel application called bilingual TC. It basically consists in the classification of bilingual documents that are translation of each other. This peculiar feature allowed us to present two classes of models (classifiers): on the one hand those models that naively assume the independence of documents conveying the same meaning, but written in different languages, and on the other hand, those models that exploit this characteristic by learning the word correlation across languages in order to improve the accuracy of their respective classifier.

For the first class of models, those assuming no direct dependency across languages, we introduced five unigram models presented in Chapter 2. For the second class of models, those learning the word mapping across languages, we proposed the unigram-M1 model in Chapter 3. For all these models we applied mixture modelling as a powerful way to deal with multimodal data.

These two classes of models were evaluated on the *Traveller* task and the *BAF* corpus, reaching the following conclusions. First, mixture-based classifiers are superior to single-component classifiers. Secondly, bilingual classifiers ourperform their monolingual counterparts on the *BAF* corpus, but this is not the case on the *Traveller* task due its simplicity. Lastly, this same conclusion can be drawn between the unigram-M1 model and the naive unigram-based models when their corresponding classification error rate figures are compared. These results were complemented with comparative experiments with other state-of-the-art learners from the field of ML, these were support vector machines and boosting techniques. These techniques offered similar performance to those presented in this thesis on the *Traveller* task, but significantly worse on the *BAF* corpus. Furthermore, Appendix A contains additional experiments to assess the performance of monolingual and bilingual text classifiers, when its feature representation goes beyond the unigram.

In Chapter 2, we also presented a real application of bilingual text classifiers in the framework of machine-aided indexing. This interesting application dealt with the automatic assignment of keywords to documents in order to describe their content. This process is performed under the supervision of a human expert that refines the output of the classification system. The satisfactory experimental results obtained on the JRC-Acquis corpus, suggested the possibility of the integration of our bilingual classifiers as the backend of a MAI system.

In statistical MT, we introduced three novel context-specific translation models as a mixture extension of the well-known M1, M2 and HMM models in Chapters 3, 4 and 5, respectively. The Viterbi alignments of these three mixture models were used to value the benefits of context-specific translation models, in the framework of two shared tasks devoted to the assessment of alignment and translation quality. The experimental results aimed at studying the alignment quality on the Hansard task reflected statistically significant reductions in alignment error rate for the M2 mixture model, showing little or no gain in the other two models. The experiments to evaluate the translation quality of the mixture models manifested minor, but systematic improvements in BLEU score for those phrase-based systems trained on a mixture model with more than one component per mixture. Moreover, the BLEU figures reported for the HMM mixture model are at the level of state-of-the-art systems on the Europarl and News-Commentary datasets.

In the case of the M2 mixture model, an iterative dynamic-programming search algorithm, designed for the conventional M2 model, was revisited in order to run additional experiments on a simple semi-artificial task. The purpose of these experiments was to analyse the evolution of the translation quality of the model under controlled experimental conditions, minimising so nuisance factors that could mask or interfere in the final results. Interestingly, the results achieved in these conditions show a statistically significant improvement in translation quality as we increase the number of components in the mixture.

In Chapter 6, we presented an interactive and predictive CAT system based on stochastic finite-state transducer technology. To this purpose, it was necessary to adapt, implement and integrate efficient error-correcting parsing and n-best translation algorithms in order to guarantee low response time while preserving adequate translations. This CAT system was automatically evaluated on two tasks, XRCE and EU corpora, revealing a significant reduction in typing effort for both tasks. An external human evaluation by translation agencies on the XRCE task, reported productivity boosts when the test texts were reasonably well correlated with the training data employed to infer the underlying stochastic finite-state transducer model.

Summarising the main contributions of this thesis are the following:

1. Bilingual TC is proposed as a novel task in text classification. We introduce four bilingual mixture models: the bilingual unigram model, the local and global decomposition models and the unigram-M1 model. We obtain good

results, being comparable or superior to those of state-of-the-art techniques on two tasks of different complexity.

2. Bilingual MAI is presented as a novel application in machine-aided indexing. The results reported on the JRC-Acquis corpus convey the possibility of a MAI system based on the models previously introduced.

3. Translation models for heterogeneous data are presented as a mixture extension of well-known word alignment translation models. More precisely, the M1, M2 and HMM mixture models are thoroughly evaluated from the viewpoint of their translation quality with minor, but systematic improvements in BLEU score; and complementary experiments to assess the alignment quality of these mixture models reflected a statistically significant reduction of AER for the M2 mixture model. The BLEU scores reported for the HMM mixture model are at the level of state-of-the-art translation systems.

4. Error-correcting and n-best parsing algorithms for stochastic finite-state transducers were adapted to work under the tight usability and low response time constraints of a CAT environment. This system was automatically and manually evaluated with satisfactory results.

7.2 Scientific publications

The content of this thesis has been published in international workshops, conferences and journals. In this section, we review those publications pointing out their relation with this thesis.

The work developed on bilingual TC was published in international conferences and workshops. More precisely, the following set of publications are related to that work in which bilingual smoothed n-gram language models were used (see appendix A):

- **J. Civera**, E. Cubel, A. Juan, and E. Vidal. Different approaches to bilingual text classification based on grammatical inference techniques. In *2nd Iberian Conference on Pattern Recognition and Image Analysis*, volume 3523 of *Lecture Notes in Computer Science*, pages 630–637. Springer-Verlag, Estoril (Portugal), June 2005.

- E. Cubel, **J. Civera**, and E. Vidal. On the use of grammatical inference techniques for bilingual text classification. In *Workshop on Grammatical Inference Applications: Successes and Future Challenges*, pages 46–50, Edinburgh (Scotland), August 2005.

- **J. Civera**, E. Cubel, and E. Vidal. Bilingual Text Classification. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 4477 of *LNCS*, pages 265–273. Springer, Girona (Spain), June 2007.

The content of Chapter 2 was published in an international workshop and an international conference:

- **J. Civera** and A. Juan. Multinomial Mixture Modelling for Bilingual Text Classification. In *Proceedings of the 6th International Workshop on Pattern Recognition in Information Systems, PRIS 2006*, pages 93–103, INSTICC Press, Paphos (Cyprus), May 2006.

- **J. Civera** and A. Juan. Bilingual Machine-Aided Indexing. In *Proceedings of the fifth international conference on Language Resources and Evaluation, LREC 2006*, pages 1302–1305, Genoa (Italy), May 2006.

The content of Chapter 3 related to the unigram-M1 mixture model for bilingual TC have been submitted to an international conference:

- **J. Civera** and A. Juan. Bilingual Text Classification using the IBM 1 Translation Model. Accepted for publication in the sixth international conference on Language Resources and Evaluation, LREC 2008.

The M2 mixture model, the extension of the dynamic-programming search algorithm and their corresponding results presented in Chapter 4 were published in an international conference:

- **J. Civera** and A. Juan. Mixtures of IBM Model 2. In *Proceedings of the 11th annual conference of the European Association for Machine Translation, EAMT 2006*, pages 159–167, Oslo (Norway), June 2006.

The HMM mixture model and some of the results presented in Chapter 5 were published in an international workshop:

- **J. Civera** and A. Juan. Domain adaptation in statistical machine translation with mixture modelling. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 177–180, Association for Computational Linguistics, Prague (Czech Republic), June 2007.

The work carried out in this thesis focused on the development of a search algorithm for interactive and predictive CAT using SFSTs (see chapter 6). This system evolved over the time in the framework of the TT2 project improving its efficiency and quality. The scientific community was timely informed of the advances of this system in numerous publications in international and national conferences:

- **J. Civera**, J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, F. Casacuberta, E. Vidal, D. Picó and J. González. A syntactic pattern recognition approach to computer assisted translation. In A. Fred, T. Caelli, A. Campilho, R. P.W. Duin, and D. de Ridder, editors, *Advances in Statistical, Structural and Syntactical Pattern Recognition*, Lecture Notes in Computer Science, pages 207–215, Springer-Verlag, Lisbon (Portugal), August 2004.

- **J. Civera**, E. Cubel, A. L. Lagarda, D. Picó, J. González, F. Casacuberta, E. Vidal, J. M. Vilar and S. Barrachina. From machine translation to computer assisted translation using finite-state models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2004*, pages 349–356, Association for Computational Linguistics, Barcelona (Spain), July 2004.

- E. Cubel, **J. Civera**, J. M. Vilar, A. L. Lagarda, F. Casacuberta, E. Vidal, D. Picó, J. González and L. Rodríguez. From machine translation to computer assisted translation using finite-state models. In *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI 2004*, pages 586–590, IOS Press, Valencia (Spain), August 2004.

- **J. Civera**, E. Cubel, A. L. Lagarda, F. Casacuberta, E. Vidal, J. M. Vilar and S. Barrachina. Computer-assisted translation using finite-state transducers. In *Actas del XXI Congreso de la Sociedad Espaola para el Procesamiento del Lenguaje Natural, SEPLN 2005*, pages 357–363, Granada (Spain), September 2005.

- **J. Civera**, J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, F. Casacuberta and E. Vidal. A novel approach to computer assisted translation based on finite-state transducers. In A. Yli-Jyra, L. Karttunen, and J. Karhumaki, editors, *Finite-State Methods and Natural Language Processing*, Lecture Notes in Artificial Intelligence (LNAI-LNCS). Springer-Verlag, Helsinki (Finland), September 2005.

- **J. Civera**, J. M. Vilar, A. L. Lagarda, E. Cubel, S. Barrachina, F. Casacuberta and E. Vidal. A Computer-Assisted Translation Tool based on Finite-State Technology. In *Proceedings of the 11th annual conference of the European Association for Machine Translation, EAMT 2006*, pages 33–40, Oslo (Norway), June 2006.

The final results of the TT2 project including those obtained with the SFST technology presented in this thesis and those of other two state-of-the-art systems based on phrase-based and alignment templates approaches, will be published in an international journal:

- S. Barrachina, O. Bender, F. Casacuberta, **J. Civera**, E. Cubel, S. Khadivi, A. L. Lagarda, H. Ney, J. Tomás, E. Vidal and J. M. Vilar. Statistical approaches to computer-assisted translation. *Computational Linguistics*, In press.

The SFST system depicted in this thesis was evaluated in real working conditions by two translation agencies that collaborated as partners in the TT2 project. A general public presentation of this system and the human evaluation performed will be published in an international journal for a very broad-based audience of computing professionals:

- F. Casacuberta, **J. Civera**, E. Cubel, A. L. Lagarda, G. Lapalme, E. Macklovitch and E. Vidal. Human interaction for high quality machine translation. *Communications of ACM*, In press.

Finally, the integration of this CAT system with an automatic speech recogniser in order to dictate translations and corrections in an interactive manner was published in an international conference and an international journal:

- L. Rodríguez, **J. Civera**, E. Vidal, F. Casacuberta, and C. Martínez. On the use of speech recognition in computer assisted translation. In *Proceedings of the InterSpeech'05*, pages 2269–2272, Lisbon (Portugal), September 2005.

- E. Vidal, F. Casacuberta, L. Rodríguez, **J. Civera**, and C. Martínez. Computer-assisted translation using speech recognition. *IEEE Transaction on Audio, Speech and Language Processing*, 14(3):941–951, 2006.

7.3 Future work

There are several research lines that would be interesting to explore as a future work in the different fields covered in this thesis.

In bilingual TC, the accuracy of the unigram mixture classifier could be significantly boosted by incorporating some of the techniques proposed in [R$^+$03, P$^+$04] from the ML community, or tackling this classification problem from the maximum entropy viewpoint [NLM99, JVN07]. An obvious continuation of the work presented is the application of mixture modelling to n-gram models [IO99] in isolation or together with a translation model. We believe that this promising line of research could return interesting results, since it combines the capability of capturing word-context and domain-specific information

In Chapter 2, we circumvented the problem of multilabel text classification training a classifier for each class independently, ignoring the overlapping among classes, and predefining the number of classes to be output. However, there have been previous works that assume the multilabel nature of the data designing specific classifiers to elegantly solve this problem [McC99, EW05, ZJXG05]. One of these approaches is the *BoosTexter* system [SS00] that was already considered in the experiments of Appendix A.

The M1 model, although proved to be useful in bilingual TC for small tasks, possesses serious limitations that make it counterproductive on large task like the JRC-Acquis corpus. This limitation is the difficulty to gather word alignment evidence in bilingual documents with excessive source or target length. For example in the JRC-Acquis corpus, it is almost impossible to learn word correlation across languages given that the average document length is over 1,500 words. A possible solution to this problem would be the derivation of a phrase-based model that integrates the M1 model at two levels. The first level would carry out the alignment of bilingual phrases defined by a segmentation hidden variable, and the second level would align words inside bilingual phrases as the usual M1 model. In this way, we would reduce the range of the alignment variable to the phrase defined in the upper-level model.

The M1 model has proved to be a versatile model that have been widely used in many different MT tasks. However, its applicability to other NLP fields is still quite unexplored. For example, cross-lingual information retrieval [PJR07], cross-lingual plagiarism and other cross-lingual application are suitable to be studied as future work.

Another interesting problem in TC is the extension of the bilingual scenario to the multilingual one. This extension is trivial in the case of the decision rules presented in Section 2.4.1 for the bilingual bag-of-words model, and the global and local decompositions. This is also the case for the naive combination of smooth n-gram models presented in Section A.1.1. In the multilingual scenario many questions arise, how can we efficiently integrate several languages into a classifier?, how would the classification error rate correlate with the inclusion of an increasing number of multilingual sources into a classifier?, how useful would be to learn word correlation across more than two languages? These are only some of the questions that multilingual TC opens for future research.

In statistical MT, the application of mixture modelling to translation models is a natural evolution of these models in the advent of larger and larger corpora with greater domain variability. Indeed, the convenience of using a weighted combination of models, instead of a single model trained on massive scale data has already been proved to be successful for large-scale language modelling in [BPX$^+$07]. In the case of language modelling, finite mixtures have been successfully explored for automatic speech recognition in [IO99], so it would be interesting to study the use of mixture of n-grams models for large-scale corpora in statistical MT. Furthermore, the derivation of other context-specific translation models, for example phrase-based or syntax-based models, or even phrase-based models parametrised by context-specific word-based translation models, are appealing and challenging issues that are worth exploring as future work.

In this thesis, finite mixture modelling has been always applied at sentence level as a continuation of the work developed on text classification, but it would be worth exploring its applicability at word level, since it directly addresses the common problem of word ambiguity in natural languages. Sentence-level and word-level mixture model context-specific p.f. in two different axis. Sentence-level mixtures consider the context defined by each sample, while word-level mixtures depend on each word to establish this context.

To conclude, the interactive and predictive approach to CAT is a promising approach, just started to be explored, with many potential users. The leverage of the statistical translation models underlying these systems is always an active research area from which CAT systems can benefit. Apart from this, the improvement in the adaptation capabilities of the CAT system to the user corrections, the incorporation of confidence measures [UN05] and the incorporation of on-line learning techniques to take full advantage of the user amendments into the CAT system, are tasks to be tackled to guarantee the usability and efficiency of interactive and predictive CAT systems.

BIBLIOGRAPHY

[BPX+07] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. Large language models in machine translation. In *Proc. of EMNLP-CoNLL'07*, pages 858–867, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[EW05] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Proc. of SIGIR'05*, pages 274–281, August 2005.

[IO99] R. M. Iyer and M. Ostendorf. Modelling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech & Audio Processing*, 7(1):30–39, 1999.

[JVN07] A. Juan, D. Vilar, and H. Ney. Bridging the gap between Naive Bayes and Maximum Entropy Text Classification. In *Proc. of PRIS'07*, pages 59–65, Funchal, Madeira - Portugal, June 2007.

[McC99] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *Proc. of AAAI'99: Workshop on Text Learning*, July 1999.

[NLM99] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proc. of IJCAI'99*, pages 61–67, July 1999.

[P+04] D. Pavlov et al. Document Preprocessing For Naive Bayes Classification and Clustering with Mixture of Multinomials. In *Proc. of KDD'04*, pages 829–834, New York, NY, USA, August 2004. ACM.

[PJR07] D. Pinto, A. Juan, and P. Rosso. Using query-relevant documents pairs for cross-lingual information retrieval. In *Proc. of TSD'07*, pages 630–637, September 2007.

[R+03] J. Rennie et al. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proc. of ICML'03*, pages 616–623, August 2003.

[SS00] R. E. Schapire and Y. Singer. Boostexter: A boosting-based systemfor text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

[UN05] N. Ueffing and H. Ney. Application of Word-Level Confidence Measures in Interactive Statistical Machine Translation. In *Proc. of EAMT'05*, pages 262–270, Budapest, Hungary, May 2005.

[ZJXG05] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *Proc. of SIGIR'05*, pages 274–281, New York, NY, USA, August 2005. ACM.

APPENDIX A

ADDITIONAL EXPERIMENTS ON BILINGUAL TEXT CLASSIFICATION

This appendix presents a series of additional experiments on bilingual TC in order to assess the performance of monolingual and bilingual text classifiers, when its feature representation goes beyond the unigram. These experiments were carried out on the same datasets introduced in Chapter 2, that is, Traveller, BAF and JRC-Acquis using smoothed n-gram language models, SVM and boosting techniques.

A.1 Experiments on Traveller and BAF datasets

A.1.1 Smoothed n-gram language models

This set of experiments was performed with the well-known and publicly available SRILM toolkit [Sto02]. The language models were trained using Witten-Bell discount [WB91] and back-off as smoothing technique. Other discount algorithms were also evaluated, but they were discarded because their performance was significantly poorer.

The general training procedure consists in generating a language model for each supervised class separately and for both languages independently. These class-dependent language models were used to define monolingual and bilingual naive Bayes classifiers. The results for the Traveller and BAF datasets are given in Table A.1 while the n-gram order ranges from unigram to trigram.

As expected, the general trend of these figures is a decrease in classification error rate as we enlarge the n-gram context window. However, bigram and trigram classifiers offer similar performance. Additional experiments demonstrated that smoothed n-gram models beyond trigrams provides no accuracy improvement at all.

Table A.1: Test-set error rates for monolingual and bilingual naive classifiers based on smoothed n-gram language models in Traveller and BAF.

Traveller	1-gram	2-gram	3-gram
English classifier	4.1	1.9	1.3
Spanish classifier	2.8	1.2	1.2
Bilingual classifier	3.3	1.2	1.1

BAF	1-gram	2-gram	3-gram
English classifier	5.3	3.5	3.6
French classifier	6.7	4.4	4.4
Bilingual classifier	4.1	2.8	2.6

Table A.2: Competing error table of the best performing bilingual classifiers on the Traveller task and the BAF corpus

	Traveller			BAF		
	1-gram	2-gram	3-gram	1-gram	2-gram	3-gram
Bilingual smoothed n-gram	3.3	1.2	1.1	4.1	2.8	2.6
Bilingual local mixture	1.4	-	-	2.9	-	-
Unigram-M1 mixture	1.3	-	-	2.5	-	-

Table A.2 shows competing errors for bilingual smoothed n-gram, bilingual local mixture and unigram-M1 mixture classifiers. The results obtained with n-gram classifiers with $n > 1$ are slightly better than the best results obtained with unigram mixtures. More precisely, the best results achieved with n-grams are 1.1% in Traveller and 2.6% in BAF, while the best results obtained with unigram mixtures are 1.4% in Traveller and 2.9% in BAF.

Furthermore, as we can observe in Table A.2, the unigram-M1 mixture model supersedes the other two unigram models proving the benefits of learning the word correlation across languages. As we move to bilingual bigrams or trigrams on the Traveller task, the context information in the same language seems to be more discriminative than the word mapping information between languages. But this is not the case on the BAF corpus, in which the M1 model is superior to n-gram models.

Nonetheless, an impartial assessment of the role of translation models when compared to bilingual smoothed n-gram ($n > 1$) should be carried out using the same underlying language models in combination to a translation model, such as the M1 model.

Table A.3: Test-set error rates on Traveller and BAF for SVM^{light} and BoosTexter

	Traveller	BAF
SVM^{light}	0.0	1.7
BoosTexter	1.0	3.9

A.1.2 Comparative results with SVM and boosting techniques

We also run comparative experiments with SVM and boosting techniques in order to explore the benefits of context information. More precisely, we studied the influence of the n-gram order and the combination of different n-gram orders, as a feature representation in SVM and boosting classifiers.

In order to tune the different parameters and the n-gram order combination up to trigrams in SVM^{light} and BoosTexter, we followed a 10-fold cross-validation on the bilingual training set. The final results for both datasets, Traveller and BAF, are shown in Table A.3

SVM^{light} offers the best results on the Traveller and BAF corpora, while BoosTexter works similarly to smoothed n-gram language models on the Traveller task and even worse on the BAF corpus.

A.2 Experiments on JRC-Acquis

A.2.1 Smoothed n-gram language models

The comparative results between smoothed n-gram (straight lines) and mixture (curves) unigram classifiers are shown in Figure A.1, both for the best monolingual (English-only) and the bilingual classifiers.

From the results in Figure A.1, we can observe that the trigram (3g) classifier performs the best on its monolingual and bilingual versions, followed by the bigram (2g) classifier, the mixture unigram (mix 1g) classifier and the unigram (1g) classifier. This performance directly correlates with the increasing length of the n-gram context window supporting these classifiers. Additional experiments demonstrated that smoothed n-gram models beyond trigrams provides no accuracy improvement at all.

A.2.2 Comparative results with boosting techniques

As we did in Section A.2.2, we carried a comparative study with boosting techniques instantiated in the toolkit BoosTexter, varying the order of the n-gram employed in the weak learner. The objective function to minimise was the Hamming loss, since ranking as learning criterion provided worse results. The best results

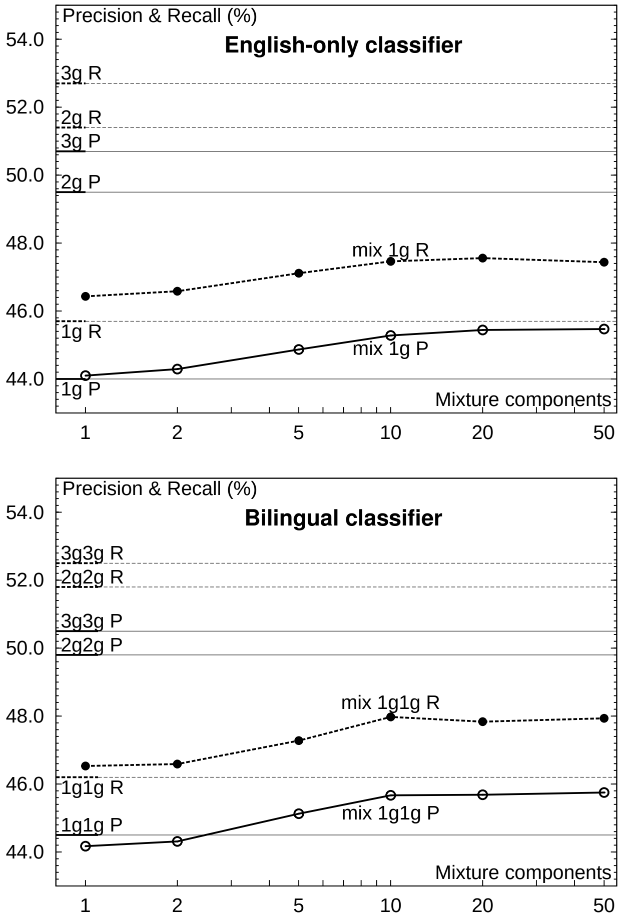


Figure A.1: Precision (P) and recall (R) curves as a function of the number of mixture components for the English-only (top) and bilingual (bottom) unigram (mix) classifiers. Precision and recall straight lines are plotted for the English-only and bilingual single component n-gram (ng) classifier.

obtained were a precision value of 44.8% and a recall value of 46.6%, that are distant from the best results using smoothed n-gram language models with a precision value of 50.7% and a recall value of 52.7%.

Finally, we evaluated the performance of a n-gram classifier as the backend of a MAI system. To this purpose, we compute the recall values that we would obtain if we considered a longer list of descriptors suggested by the system. The results were that the system would be offering up to 69.1% of the correct descriptors for a list of 10 descriptors. This value can be considered fairly good considering our MAI system as an external annotator that always returns 10 descriptors and given that the human annotator agreement is between 70%-80%.

BIBLIOGRAPHY

[Sto02] A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proc. of ICSLP'02*, pages 901–904, September 2002.

[WB91] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37:1085–1094, 1991.

APPENDIX B
EM DERIVATION

This appendix assumes that the reader is familiar with the chapters in which these models were initially presented. The objective of this appendix is to provide further details of the derivation of the models introduced throughout this thesis.

However, the EM algorithm for the bilingual unigram, unigram-M1 and M2 models is not included in this appendix due their similar derivation to the unigram and M1 models that are presented in this appendix.

B.1 EM algorithm for finite mixture models

This section depicts the E and M steps of a finite mixture model that are thoroughly used in Chapters 2, 3, 4 and 5.

B.1.1 E step for finite mixture models

As seen in Eq. (1.44), we need to compute the expected value of the indicator variable z_{nt}. This variable is 1 if the nth sample was generated by the tth component of the mixture and 0 otherwise. Thus,

$$\begin{aligned} z_{nt}^{(k)} &= E(z_{nt} \,|\, x_n; \mathbf{\Theta}^{(k)}) \\ &= \sum_{z_{nt}} z_{nt}\, p(z_{nt} \,|\, x_n; \mathbf{\Theta}^{(k)}) \\ &= p(z_{nt} = 1 \,|\, x_n; \mathbf{\Theta}^{(k)}). \end{aligned} \tag{B.1}$$

So, the expected value of the variable z_{nt} in the kth iteration is the posterior probability of the tth being responsible for the generation of the nth sample given the current estimation of $\mathbf{\Theta}$. This posterior is computed as

$$z_{nt}^{(k)} = \frac{p(z_{nt}=1)\, p(x_n \,|\, z_{nt}=1; \mathbf{\Theta}_t^{(k)})}{\sum_{t'=1}^{T} p(z_{nt'}=1)\, p(x_n \,|\, z_{nt'}=1; \mathbf{\Theta}_{t'}^{(k)})} \tag{B.2}$$

$$= \frac{p(t)^{(k)}\, p(x_n \,|\, t; \mathbf{\Theta}_t^{(k)})}{\sum_{t'=1}^{T} p(t')^{(k)}\; p(x_n \,|\, t'; \mathbf{\Theta}_{t'}^{(k)})} \tag{B.3}$$

B.1.2 M step for finite mixture models

In the M step, we maximise the Q function along with its associated Lagrange multipliers as in Eq. (1.46). This maximisation entails taking derivatives of Eq. (1.46) w.r.t. $\mathbf{\Theta}$ and the associated Lagrange multipliers, and equating them to zero.

In the case of mixture coefficients, we take derivatives w.r.t. $p(t)$ and its corresponding Lagrange multiplier λ equating to zero

$$\frac{\partial\, Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) + \Lambda}{\partial\, p(t)} = \sum_{n=1}^{N} \frac{z_{nt}^{(k)}}{p(t)} - \lambda = 0 \tag{B.4}$$

$$\frac{\partial\, Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) + \Lambda}{\partial\, \lambda} = \sum_{t=1}^{T} p(t) - 1 = 0. \tag{B.5}$$

Reorganising these equations, so we can substitute one into the other

$$p(t) = \sum_{n=1}^{N} \frac{z_{nt}^{(k)}}{\lambda} \tag{B.6}$$

$$\sum_{t=1}^{T} p(t) = 1 \tag{B.7}$$

then,

$$\sum_{t=1}^{T} \sum_{n=1}^{N} \frac{z_{nt}^{(k)}}{\lambda} = 1 \tag{B.8}$$

where

$$\lambda = \sum_{t=1}^{T} \sum_{n=1}^{N} z_{nt}^{(k)} \tag{B.9}$$

and replacing λ into Eq. (B.6), we have

$$p(t)^{(k+1)} = \frac{\sum\limits_{n=1}^{N} z_{nt}^{(k)}}{\sum\limits_{t'=1}^{T} \sum\limits_{n=1}^{N} z_{nt'}^{(k)}}$$

$$= \frac{1}{N} \sum_{n=1}^{N} z_{nt}^{(k)}. \tag{B.10}$$

For the component-conditional parameters, we take derivatives w.r.t. $\mathbf{\Theta}_t$ and its Lagrange multiplier (if any) equating to zero

$$\frac{\partial\, Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) + \Lambda}{\partial\, \mathbf{\Theta}_t} = 0 \tag{B.11}$$

B.2 EM algorithm for the unigram mixture model

This section aims at clarifying the EM algorithm applied to the derivation of the unigram model presented in Section 2.2.2. This derivation is similar to that of Section B.1, considering the component-conditional p.f. to be a unigram model. Indeed, the derivation of the E step for unigram mixture model is the same to that presented in Section B.1.1, and therefore it is omitted.

B.2.1 M step for the unigram mixture model

In the M step, we maximise Eq. (1.46), with an additional constraint

$$\sum_{u \in \mathcal{X}} p(u \,|\, t) = 1 \quad \forall\, t. \tag{B.12}$$

to normalise the unigram parameters. So, we redefine Λ in Eq. (1.46) as,

$$\Lambda = \begin{cases} -\lambda \left(\sum\limits_{t=1}^{T} p(t) - 1 \right) \\ -\sum\limits_{t=1}^{T} \mu_t \left(\sum\limits_{u \in \mathcal{X}} p(u \,|\, t) - 1 \right) . \end{cases} \tag{B.13}$$

Now we can take derivatives of Eq. (1.46) w.r.t. $\mathbf{\Theta}$ and Λ equating to zero. The derivation of the mixture coefficients was already presented in Section B.2.1, so here, we focus on the derivation of $p(u \,|\, t)$. First, we take derivatives and equate to zero

$$\frac{\partial Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) + \Lambda}{\partial p(u \,|\, t)} = \sum_{n=1}^{N} \sum_{j: x_{nj} = u}^{|x_n|} \frac{z_{nt}^{(k)}}{p(u \,|\, t)} - \mu_t = 0 \tag{B.14}$$

$$\frac{\partial Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) + \Lambda}{\partial \mu_t} = \sum_{u \in \mathcal{X}} p(u \,|\, t) - 1 = 0 \tag{B.15}$$

Reorganising these equations, so we can substitute one into the other

$$p(u \,|\, t) = \sum_{n=1}^{N} \sum_{j: x_{nj} = u}^{|x_n|} \frac{z_{nt}^{(k)}}{\mu_t} \tag{B.16}$$

$$\sum_{u \in \mathcal{X}} p(u \,|\, t) = 1 \tag{B.17}$$

then,

$$\sum_{u\in\mathcal{X}}\sum_{n=1}^{N}\sum_{j:x_{nj}=u}^{|x_n|}\frac{z_{nt}^{(k)}}{\mu_t}=1 \tag{B.18}$$

where

$$\mu_t=\sum_{u\in\mathcal{X}}\sum_{n=1}^{N}\sum_{j:x_{nj}=u}^{|x_n|}z_{nt}^{(k)} \tag{B.19}$$

and replacing μ_t into Eq. (B.16), we have

$$p(u\,|\,t)^{(k+1)}=\frac{\sum\limits_{n=1}^{N}\sum\limits_{j:x_{nj}=u}^{|x_n|}z_{nt}^{(k)}}{\sum\limits_{u'\in\mathcal{X}}\sum\limits_{n=1}^{N}\sum\limits_{j:x_{nj}=u'}^{|x_n|}z_{nt}^{(k)}} \tag{B.20}$$

B.3 EM algorithm for the M1 model

This section provides details of the derivation of E and M steps for the M1 model. The E step of the M1 mixture model is briefly described in order to clarify the computation of the expected value of the product of two hidden variables and its notation. The derivation of the M step of the M1 mixture model is straightforward provided that the updating equation for the mixture coefficients is the same to that in Section B.1.2, and the derivation of the updating equation of the component-dependent statistical dictionary is analogous to that of the conventional statistical dictionary. So, we omit the derivation of the M step of the M1 mixture model.

B.3.1 E step in the M1 model

In the M1 model, the E step reduces to compute the expected value of the indicator variable a_{nji}, as seen in Eq. (3.14). This variable is 1 if there is an alignment between the jth source position to the ith target position in the nth sample and 0 otherwise.

$$\begin{aligned}a_{nji}^{(k)}&=E(a_{nji}\,|\,x_n,y_n;\mathbf{\Theta}^{(k)})\\&=\sum_{a_{nji}}a_{nji}\,p(a_{nji}\,|\,x_n,y_n;\mathbf{\Theta}^{(k)})\\&=p(a_{nji}=1\,|\,x_n,y_n;\mathbf{\Theta}^{(k)}).\end{aligned} \tag{B.21}$$

So, the expected value of a_{nji} in the kth iteration is the posterior probability of the source position j to be connected to the target position i given the source and target sentence and the current estimation of $\boldsymbol{\Theta}$. This posterior is computed as follows

$$\begin{aligned}
a_{nji}^{(k)} &= \frac{p(x_n, a_{nji} = 1 \,|\, y_n; \boldsymbol{\Theta}^{(k)})}{\sum\limits_{i'=0}^{|y_n|} p(x_n, a_{nji'} = 1 \,|\, y_n; \boldsymbol{\Theta}^{(k)})} \\
&= \frac{p(x_{nj}, a_{nji} = 1 \,|\, y_n; \boldsymbol{\Theta}^{(k)})\, p(x_{n1}^{j-1}, x_{nj+1}^{|x_n|} \,|\, y_n; \boldsymbol{\Theta}^{(k)})}{\sum\limits_{i'=0}^{|y_n|} p(x_{nj}, a_{nji'} = 1 \,|\, y_n; \boldsymbol{\Theta}^{(k)})\, p(x_{n1}^{j-1}, x_{nj+1}^{|x_n|} \,|\, y_n; \boldsymbol{\Theta}^{(k)})} \\
&= \frac{p(a_{nji} = 1 \,|\, y_n; \boldsymbol{\Theta}^{(k)})\, p(x_{nj} \,|\, a_{nji} = 1, y_n; \boldsymbol{\Theta}^{(k)})}{\sum\limits_{i'=0}^{|y_n|} p(a_{nji'} = 1 \,|\, y_n; \boldsymbol{\Theta}^{(k)})\, p(x_{nj} \,|\, a_{nji'} = 1, y_n; \boldsymbol{\Theta}^{(k)})} \\
&= \frac{p(x_{nj} \,|\, y_{ni})^{(k)}}{\sum\limits_{i'=0}^{|y_n|} p(x_{nj} \,|\, y_{ni'})^{(k)}}. \qquad \text{(B.22)}
\end{aligned}$$

B.3.2 M step in the M1 model

In the M step, we maximise the function Q in Eq. (3.14), with the constraint that the dictionary probabilities sum up to 1

$$\sum_{u \in \mathcal{X}} p(u \,|\, v) = 1 \quad \forall\, v. \qquad \text{(B.23)}$$

As in Eq. (1.46), we incorporate this contraint with Lagrange multipliers

$$\Lambda = -\sum_{v \in \mathcal{Y}} \lambda_v \left(\sum_{u \in \mathcal{X}} p(u \,|\, v) - 1 \right). \qquad \text{(B.24)}$$

Now we can take derivatives of Eq. (1.46) w.r.t $p(u \,|\, v)$ and λ_v equating to zero

$$\frac{\partial\, Q(\boldsymbol{\Theta} \,|\, \boldsymbol{\Theta}^{(k)}) - \Lambda}{\partial\, p(u \,|\, v)} = \sum_{n=1}^{N} \sum_{j: x_{nj} = u}^{|x_n|} \sum_{i: y_{ni} = v}^{|y_n|} \frac{a_{nji}^{(k)}}{p(u \,|\, v)} - \lambda_v = 0 \qquad \text{(B.25)}$$

$$\frac{\partial\, Q(\boldsymbol{\Theta} \,|\, \boldsymbol{\Theta}^{(k)}) - \Lambda}{\partial\, \lambda_v} = \sum_{u \in \mathcal{X}} p(u \,|\, v) - 1 = 0. \qquad \text{(B.26)}$$

Reorganising both derivatives

$$p(u \mid v) = \sum_{n=1}^{N} \sum_{j:x_{nj}=u}^{|x_n|} \sum_{i:y_{ni}=v}^{|y_n|} \frac{a_{nji}^{(k)}}{\lambda_v} \tag{B.27}$$

$$\sum_{u \in \mathcal{X}} p(u \mid v) = 1. \tag{B.28}$$

and plugging Eq. (B.27) into Eq. (B.28)

$$\sum_{u \in \mathcal{X}} \sum_{n=1}^{N} \sum_{j:x_{nj}=u}^{|x_n|} \sum_{i:y_{ni}=v}^{|y_n|} \frac{a_{nji}^{(k)}}{\lambda_v} = 1 \tag{B.29}$$

where

$$\lambda_v = \sum_{u \in \mathcal{X}} \sum_{n=1}^{N} \sum_{j:x_{nj}=u}^{|x_n|} \sum_{i:y_{ni}=v}^{|y_n|} a_{nji}^{(k)}. \tag{B.30}$$

Therefore, substituting Eq. (B.30) into Eq. (B.27), we have the update equation for the statistical dictionary in the M1 model

$$p(u \mid v)^{(k+1)} = \frac{\sum\limits_{n=1}^{N} \sum\limits_{j:x_{nj}=u}^{|x_n|} \sum\limits_{i:y_{ni}=v}^{|y_n|} a_{nji}^{(k)}}{\sum\limits_{u' \in \mathcal{X}} \sum\limits_{n=1}^{N} \sum\limits_{j:x_{nj}=u'}^{|x_n|} \sum\limits_{i:y_{ni}=v}^{|y_n|} a_{nji}^{(k)}.} \tag{B.31}$$

B.3.3 E step for the M1 mixture model

In Eq. (3.24), we need to compute $z_{nt}^{(k)}$ and $(z_{nt}\, a_{nji})^{(k)}$. On the one hand, $z_{nt}^{(k)}$ is calculated as a y-conditional version of that in Eq. (B.1). On the other hand, the computation of $(z_{nt}\, a_{nji})^{(k)}$ is simplified taking into account that their product is different from zero when both variable are evaluated to one

$$\begin{aligned} (z_{nt}\, a_{nji})^{(k)} &= E(z_{nt}\, a_{nji} \mid x_n, y_n; \boldsymbol{\Theta}^{(k)}) \\ &= \sum_{z_{nt}\, a_{nji}} z_{nt}\, a_{nji}\, p(z_{nt}, a_{nji} \mid x_n, y_n; \boldsymbol{\Theta}^{(k)}) \\ &= p(z_{nt} = 1, a_{nji} = 1 \mid x_n, y_n; \boldsymbol{\Theta}^{(k)}) \\ &= z_{nt}^{(k)}\, a_{njit}^{(k)}. \end{aligned} \tag{B.32}$$

as seen in Eq. (3.26), therefore the computation of $a_{njit}^{(k)}$ is a component-conditional version of $a_{nji}^{(k)}$ in Eq. (B.22)

$$
\begin{aligned}
a_{njit}^{(k)} &= \frac{p(x_n, a_{nji} = 1 \,|\, y_n, z_{nt} = 1; \mathbf{\Theta}^{(k)})}{\sum\limits_{i'=0}^{|y_n|} p(x_n, a_{nji'} = 1 \,|\, y_n, z_{nt} = 1; \mathbf{\Theta}^{(k)})} \\
&= \frac{p(a_{nji} = 1 \,|\, y_n, z_{nt} = 1; \mathbf{\Theta}^{(k)})\, p(x_{nj} \,|\, a_{nji} = 1, y_n, z_{nt} = 1; \mathbf{\Theta}^{(k)})}{\sum\limits_{i'=0}^{|y_n|} p(a_{nji'} = 1 \,|\, y_n, z_{nt} = 1; \mathbf{\Theta}^{(k)})\, p(x_{nj} \,|\, a_{nji'} = 1, y_n, z_{nt} = 1; \mathbf{\Theta}^{(k)})} \\
&= \frac{p(x_{nj} \,|\, y_{ni}, t)^{(k)}}{\sum\limits_{i'=0}^{|y_n|} p(x_{nj} \,|\, y_{ni'}, t)^{(k)}} \qquad \text{(B.33)}
\end{aligned}
$$

B.4 EM algorithm for the HMM model

This section carries out a presentation of the E and M step for the HMM alignment model in a similar fashion to the M1 model in Section B.3. Also it depicts the computation of the E-step of the HMM mixture model, and as we did in Section B.3, it omits the derivation of the M step of the HMM mixture model, since it is considered to be straightforward given the derivation of the M step in Section (B.4.2).

B.4.1 E step in the HMM model

In the HMM model, the E step in Eq. (5.15) requires the computation of the expected value of the indicator variable a_{nji} for the alignment of the first position and for the statistical dictionary, and the product $a_{nj-1i'}\, a_{nji}$ for the jump width alignment parameter.

The computation of a_{nji} and $a_{nj-1i'}\, a_{nji}$ is somewhat more complex given the first-order dependency of the HMM model and requires the usage of the α and β

recursive functions in Eqs. (5.9) and (5.10). The term $a_{nji}^{(t)}$ is calculated as

$$
\begin{aligned}
a_{nji}^{(k)} &= p(a_{nji} = 1 \,|\, x_n, y_n; \mathbf{\Theta}^{(k)}) \\
&= \frac{p(x_n, a_{nji} = 1 \,|\, y_n; \mathbf{\Theta}^{(k)})}{\sum_{\tilde{\imath}=1}^{|y_n|} p(x_n, a_{nj\tilde{\imath}} = 1 \,|\, y_n; \mathbf{\Theta}^{(k)})} \\
&= \frac{p(x_{n1}^{j}, a_{nji} = 1 \,|\, y_n; \mathbf{\Theta}^{(k)})\, p(x_{nj+1}^{|x_n|} \,|\, x_{n1}^{j}, a_{nji} = 1, y_n; \mathbf{\Theta}^{(k)})}{\sum_{\tilde{\imath}=1}^{|y_n|} p(x_{n1}^{j}, a_{nj\tilde{\imath}} = 1 \,|\, y_n; \mathbf{\Theta}^{(k)})\, p(x_{nj+1}^{|x_n|} \,|\, x_{n1}^{j}, a_{nj\tilde{\imath}} = 1, y_n; \mathbf{\Theta}^{(k)})} \\
&= \frac{\alpha_{nji}\beta_{nji}}{\sum_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj\tilde{\imath}}\beta_{nj\tilde{\imath}}}
\end{aligned}
\tag{B.34}
$$

and $(a_{nj-1i'}\, a_{nji})^{(k)}$ as

$$
\begin{aligned}
(a_{nj-1i'} a_{nji})^{(k)} &= p(a_{nj-1i'} = 1, a_{nji} = 1 \,|\, x_n, y_n; \mathbf{\Theta}^{(k)}) \\
&= \frac{p(x_n, a_{nj-1i'} = 1, a_{nji} = 1 \,|\, y_n; \mathbf{\Theta}^{(k)})}{\sum_{\tilde{\imath}'=1}^{|y_n|} \sum_{\tilde{\imath}=1}^{|y_n|} p(x_n, a_{nj-1\tilde{\imath}'} = 1, a_{nj\tilde{\imath}} = 1 \,|\, y_n; \mathbf{\Theta}^{(k)})}
\end{aligned}
\tag{B.35}
$$

where

$$
\begin{aligned}
p(x_n, a_{nj-1i'} = 1, a_{nji} = 1 \,|\, y_n; \mathbf{\Theta}^{(k)}) = {} & p(x_{n1}^{j-1}, a_{nj-1i'} = 1 \,|\, y_n; \mathbf{\Theta}^{(k)}) \\
& p(a_{nji} = 1 \,|\, x_{n1}^{j-1}, a_{nj-1i'} = 1, y_n; \mathbf{\Theta}^{(k)}) \\
& p(x_{nj} \,|\, x_{n1}^{j-1}, a_{nj-1i'} = 1, a_{nji} = 1, y_n; \mathbf{\Theta}^{(k)}) \\
& p(x_{nj+1}^{|x_n|} \,|\, x_{n1}^{j}, a_{nj-1i'} = 1, a_{nji} = 1, y_n; \mathbf{\Theta}^{(k)}) \\
= {} & \alpha_{nj-1i'}\, p(i \,|\, i', |y|)^{(k)}\, p(x_{nj}|y_{ni})^{(k)}\, \beta_{nji}.
\end{aligned}
\tag{B.36}
$$

Then,

$$
(a_{nj-1i'} a_{nji})^{(k)} = \frac{\alpha_{nj-1i'}\, p(i \,|\, i', |y|)^{(k)}\, p(x_{nj}|y_{ni})^{(k)}\, \beta_{nji}}{\sum_{\tilde{\imath}'=1}^{|y_n|} \sum_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj-1\tilde{\imath}'}\, p(\tilde{\imath} \,|\, \tilde{\imath}', |y|)^{(k)}\, p(x_{nj}|y_{n\tilde{\imath}})^{(k)}\, \beta_{nj\tilde{\imath}}}
\tag{B.37}
$$

B.4.2 M step in the HMM model

In the M step, we maximise the Q function in Eq. (5.6), under the following constraints

$$\begin{array}{ll} \sum_{i=1}^{|y|} p(i \,|\, i', |y|) = 1 & \forall\, 1 \leq i' \leq |y| \text{ and } |y| \\ \sum_{u \in \mathcal{X}} p(u \,|\, v) = 1 & \forall\, v \end{array} \tag{B.38}$$

that are translated into Lagrange multipliers as

$$\Lambda = \begin{cases} -\sum_{i'=1}^{|y|} \sum_{|y|} \lambda_{i'|y|} \left(\sum_{i=1}^{|y|} p(i \,|\, i', |y|) - 1 \right) \\ -\sum_{v \in \mathcal{Y}} \mu_v \left(\sum_{u \in \mathcal{X}} p(u \,|\, v) - 1 \right) \end{cases} \tag{B.39}$$

in Eq. (1.46). Then, we can take derivatives of Eq. (1.46) w.r.t. $\mathbf{\Theta}$ and Λ equating to zero. Starting with $p(i \,|\, i', |y|)$, we take derivatives and equate to zero w.r.t. this parameter and its corresponding Langrange multiplier

$$\frac{\partial Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) + \Lambda}{\partial p(i \,|\, i', |y|)} = \sum_{\substack{n=1 \\ |y_n|=|y|}}^{N} \sum_{j=1}^{|x_n|} \frac{(a_{nj-1i'}\, a_{nji})^{(k)}}{p(i \,|\, i', |y|)} - \lambda_{i'|y|} = 0 \tag{B.40}$$

$$\frac{\partial Q(\mathbf{\Theta} \,|\, \mathbf{\Theta}^{(k)}) + \Lambda}{\partial \lambda_{i'|y|}} = \sum_{i=1}^{|y|} p(i \,|\, i', |y|) - 1 = 0 \tag{B.41}$$

so

$$p(i \,|\, i', |y|) = \sum_{\substack{n=1 \\ |y_n|=|y|}}^{N} \sum_{j=1}^{|x_n|} \frac{(a_{nj-1i'}\, a_{nji})^{(k)}}{\lambda_{i'|y|}} \tag{B.42}$$

$$\sum_{i=1}^{|y|} p(i \,|\, i', |y|) = 1 \tag{B.43}$$

then,

$$\sum_{i=1}^{|y|} \sum_{\substack{n=1 \\ |y_n|=|y|}}^{N} \sum_{j=1}^{|x_n|} \frac{(a_{nj-1i'}\, a_{nji})^{(k)}}{\lambda_{i'|y|}} = 1 \tag{B.44}$$

where

$$\lambda_{i'|y|} = \sum_{i=1}^{|y|} \sum_{\substack{n=1 \\ |y_n|=|y|}}^{N} \sum_{j=1}^{|x_n|} (a_{nj-1i'}\, a_{nji})^{(k)} \tag{B.45}$$

and substituting $\lambda_{i'|y|}$ into Eq. (B.42), we have

$$p(i \mid i', |y|)^{(k+1)} = \frac{\sum\limits_{\substack{n=1 \\ |y_n|=|y|}}^{N} \sum\limits_{j=1}^{|x_n|} (a_{nj-1i'}\, a_{nji})^{(k)}}{\sum\limits_{\tilde{\imath}=1}^{|y|} \sum\limits_{\substack{n=1 \\ |y_n|=|y|}}^{N} \sum\limits_{j=1}^{|x_n|} (a_{nj-1i'}\, a_{nj\tilde{\imath}})^{(k)}}. \tag{B.46}$$

Finally, for $p(u \mid v)$ we follow the same procedure

$$\frac{\partial Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)}) + \Lambda}{\partial p(u \mid v)} = \sum_{n=1}^{N} \sum_{j:x_{nj}=u}^{|x_n|} \sum_{i:y_{ni}=v}^{|y_n|} \frac{a_{nji}^{(k)}}{p(u \mid v)} - \lambda_v = 0 \tag{B.47}$$

$$\frac{\partial Q(\boldsymbol{\Theta} \mid \boldsymbol{\Theta}^{(k)}) + \Lambda}{\partial \lambda_v} = \sum_{u \in \mathcal{X}} p(u \mid v) - 1 = 0. \tag{B.48}$$

Reorganising terms

$$p(u \mid v) = \sum_{n=1}^{N} \sum_{j:x_{nj}=u}^{|x_n|} \sum_{i:y_{ni}=v}^{|y_n|} \frac{a_{nji}^{(k)}}{\lambda_v} \tag{B.49}$$

$$\sum_{u \in \mathcal{X}} p(u \mid v) = 1 \tag{B.50}$$

then,

$$\sum_{u \in \mathcal{X}} \sum_{n=1}^{N} \sum_{j:x_{nj}=u}^{|x_n|} \sum_{i:y_{ni}=v}^{|y_n|} \frac{a_{nji}^{(k)}}{\lambda_v} = 1 \tag{B.51}$$

where

$$\lambda_v = \sum_{u \in \mathcal{X}} \sum_{n=1}^{N} \sum_{j:x_{nj}=u}^{|x_n|} \sum_{i:y_{ni}=v}^{|y_n|} a_{nji}^{(k)}. \tag{B.52}$$

As we did before, replacing λ_v into Eq. (B.49)

$$p(u \mid v)^{(k+1)} = \frac{\sum\limits_{n=1}^{N} \sum\limits_{j:x_{nj}=u}^{|x_n|} \sum\limits_{i:y_{ni}=v}^{|y_n|} a_{nji}^{(k)}}{\sum\limits_{u' \in \mathcal{X}} \sum\limits_{n=1}^{N} \sum\limits_{j:x_{nj}=u'}^{|x_n|} \sum\limits_{i:y_{ni}=v}^{|y_n|} a_{nji}^{(k)}.} \tag{B.53}$$

B.4.3 E step for the HMM mixture model

As seen in Eq. (5.15), we need to compute $z_{nt}^{(k)}$, $(z_{nt}\, a_{nji})^{(k)}$ and $(z_{nt}\, a_{nj-1i'}\, a_{nji})^{(k)}$. First, $z_{nt}^{(k)}$ is computed in the usual way considering a y-conditional version of

Eq (B.1), taking component-dependent HMM alignment models as component-dependent p.f. in a finite mixture.

The term $(z_{nt}\, a_{nji})^{(k)}$ is decomposed as in Eq. (3.26), and the resultant term $a_{njit}^{(k)}$ can be computed as a component-dependent version of Eq. (B.34)

$$\begin{aligned} a_{njit}^{(k)} &= p(a_{nji} = 1 \,|\, x_n, z_{nt} = 1, y_n; \boldsymbol{\Theta}^{(k)}) \\ &= \frac{p(x_n, a_{nji} = 1 \,|\, z_{nt} = 1, y_n; \boldsymbol{\Theta}^{(k)})}{\sum_{\tilde{\imath}=1}^{|y_n|} p(x_n, a_{nj\tilde{\imath}} = 1 \,|\, z_{nt} = 1, y_n; \boldsymbol{\Theta}^{(k)})} \\ &= \frac{\alpha_{njit}\, \beta_{njit}}{\sum_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj\tilde{\imath}t}\, \beta_{nj\tilde{\imath}t}} . \end{aligned} \tag{B.54}$$

Finally, the term $(z_{nt}\, a_{nj-1i'}\, a_{nji})^{(k)}$ is decomposed as we did in Eq. (5.17), where $(a_{nj-1i'}\, a_{nji} \,|\, t)^{(k)}$ is computed as a component-dependent version of Eq. (B.35)

$$\begin{aligned} &(a_{nj-1i'}\, a_{nji} \,|\, t)^{(k)} = \\ &\qquad = p(a_{nj-1i'} = 1, a_{nji} = 1 \,|\, z_{nt} = 1, x_n, y_n; \boldsymbol{\Theta}^{(k)}) \\ &\qquad = \frac{p(x_n, a_{nj-1i'} = 1, a_{nji} = 1 \,|\, z_{nt} = 1, y_n; \boldsymbol{\Theta}^{(k)})}{\sum_{\tilde{\imath}'=1}^{|y_n|} \sum_{\tilde{\imath}=1}^{|y_n|} p(x_n, a_{nj-1\tilde{\imath}'} = 1, a_{nj\tilde{\imath}} = 1 \,|\, z_{nt} = 1, y_n; \boldsymbol{\Theta}^{(k)})} \\ &\qquad = \frac{\alpha_{nj-1i't}\, p(i \,|\, i', |y|, t)^{(k)}\, p(x_{nj}|y_{ni}, t)^{(k)}\, \beta_{njit}}{\sum_{\tilde{\imath}'=1}^{|y_n|} \sum_{\tilde{\imath}=1}^{|y_n|} \alpha_{nj-1\tilde{\imath}'t}\, p(\tilde{\imath} \,|\, \tilde{\imath}', |y|, t)^{(k)}\, p(x_{nj}|y_{n\tilde{\imath}}, t)^{(k)}\, \beta_{nj\tilde{\imath}t}} . \end{aligned} \tag{B.55}$$

Appendix C

Symbols and acronyms

C.1 Mathematical symbols

$\lvert\cdot\rvert$	cardinal of a set or word sequence length.
$a = a_1^{\lvert x\rvert} = a_1, \dots, a_j, \dots, a_{\lvert x\rvert}$	alignment sequence.
a_j	target position to which is aligned the jth source position.
$\boldsymbol{a}_j$	indicator vector for alignment of jth source position.
a_{ji}	indicator variable for alignment of jth source position to ith target position.
$\mathcal{A}(x, y)$	set of all possible alignments from x to y
$A = a_1, \dots, a_n, \dots, a_N$	vector of alignments.
c	class label.
$C = c_1, \dots, c_n, \dots, c_N$	vector of class labels.
j	index for the source sentence.
ĩ	secondary index for the target sentence.
(k)	iteration of EM algorithm.
n	index for a set of samples.
N	number of samples.
$p(\cdot)$	general probability function.
$p(\cdot)$	model probability distribution.
t	index for components in a mixture model.
T	number of components in a mixture model.
u	word in a source language.
v	word in a target language.
x_j	jth word in a source sentence.
$x = x_1^{\lvert x\rvert} = x_1, \dots, x_j, \dots, x_{\lvert x\rvert}$	sequence of source words.
$\overline{x}$	source segment or phrase.
$\mathcal{X}$	source vocabulary.
$X = x_1, \dots, x_n, \dots, x_N$	vector of source sentences.

y_i	ith word in a target language.
$y = y_1^{\lvert y\rvert} = y_1, \dots, y_i, \dots, y_{\lvert y\rvert}$	sequence of target words.
$\overline{y}$	target segment or phrase.
$\mathcal{Y}$	target vocabulary.
$Y = y_1, \dots, y_n, \dots, y_N$	vector of target sentences.
$\boldsymbol{z} = z_1, \dots, z_t, \dots, z_T$	indicator vector for mixture components.
$Z = \boldsymbol{z}_1, \dots, \boldsymbol{z}_n, \dots, \boldsymbol{z}_N$	vector of indicator vectors.
ϵ	interpolation parameter for uniform smooth.
$\boldsymbol{\Theta}$	parameter vector for a model.
$\boldsymbol{\Psi}$	parameter vector for a set of classes.

C.2 Acronyms

p.f.	probability function
w.r.t.	with respect to
ACL	association for computational linguistics
AER	alignment error rate
BLEU	bilingual evaluation understudy
BP	brevity penalty
BAF	bitextes anglais-français
CAT	computer-assisted translation
CER	classification error rate
CER	character error rate
GIATI	grammatical inference and alignments for transducer inference
EM	expectation maximisation
EU	European Union
HMM	hidden Markov model
IBM	international business machines
KSR	key-stroke ratio
KMSR	key-stroke and mouse-action ratio
LCP	longest common character prefix
MAI	machine-aided indexing
MAR	mouse-action ration
MERT	minimum error rate training
ML	machine learning
MT	machine translation
NLP	natural language processing
REA	recursive enumeration algorithm
SER	sentence error rate
SFST	stochastic finite-state transducers
SRILM	Stanford research institute language modeling
SVM	support vector machines
TER	translation edit rate
UN	United Nations
WER	word error rate

LIST OF FIGURES

LIST OF TABLES