UNIVERSITAT POLITÈCNICA DE VALÈNCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Building task-oriented machine translation systems

Ph. D. Thesis
presented by Germán Sanchis Trilles
supervised by Prof. Francisco Casacuberta Nolla

June 20, 2012

Building task-oriented machine translation systems

Germán Sanchis Trilles

Thesis performed under the supervision of doctor
Francisco Casacuberta Nolla
and presented at the Universitat Politècnica de València
in partial fulfilment of the requirements
for the degree Doctor en Informática

Valencia, June 20, 2012

Work supported by the EC (FP7) under CasMaCat (287576) project and also by the EC (FEDER/FSE) and the Spanish MICINN under projects MIPRCV "Consolider Ingenio 2010" (CSD2007-00018) and iTrans2 (TIN2009-14511). Also supported by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project, by the Spanish MEC under grant TIN2006-15694-CO2-01, by the Generalitat Valenciana under grant Prometeo/2009/014, and by the UPV under grant 20091027.

A mis padres.
A los invariantes.

Abstract

The main goal of this thesis is to develop computer assisted translation and machine translation systems which present a more robust synergy with their potential users. Hence, the main purpose is to make current state-of-the-art systems more ergonomic, intuitive and efficient, so that the human expert feels more comfortable when using them. For doing this, different techniques are presented, focusing on improving the adaptability and response time of the underlying statistical machine translation systems, as well as a strategy aiming at enhancing human-machine interaction within an interactive machine translation setup. All of this with the ultimate purpose of filling in the existing gap between the state of the art in machine translation and the final tools that are usually available for the final human translators.

Concerning the response time of the machine translation systems, a parameter pruning technique is presented, whose intuition stems from the concept of bilingual segmentation, but which evolves towards a full parameter re-estimation strategy. By using such strategy, experimental results presented here prove that it is possible to achieve reductions of up to 97% in the number of parameters required without a significant loss in translation quality. Being robust across different language pairs, these results evidence that the pruning technique presented is effective in a traditional machine translation scenario, and could be used for instance in a post-editing setup. Nevertheless, experiments carried out within a simulated interactive machine translation environment are slightly less convincing, since a trade-off between response time and translation quality is needed.

Two orthogonally different approaches are presented with the purpose of increasing the adaptability of the statistical machine translation systems. On the one hand, we investigate how to increase the adaptability of the language model, by subdividing it into several smaller language models which are then interpolated in translation time according to the source sentence to be translated. The specific sub-models are built either by taking advantage of supervised information present in certain bilingual corpora, or by performing unsupervised clustering on the training set, with the aim of uncovering specific sub-topics or language

styles present. On the other hand, Bayesian predictive adaptation is elucidated as an efficient strategy for adapting the translation models present in state-of-the-art machine translation systems. Although adaptation experiments are only performed within the traditional machine translation framework, the results obtained are compelling enough for implementing them within an interactive setup, and such work will be done in the near future. Nevertheless, it should be noted that the techniques developed may be readily implemented within a computer assisted translation scenario, in which a statistical machine translation system is providing the translations that the user needs to modify and validate.

Finally, special attention is devoted to increasing the synergy between the human expert and the interactive machine translation system. With this purpose, two different forms of weaker feedback are studied, which intend to increase the productivity of the human translator. For doing this, two different changes to the traditional interaction scheme are presented. The first one aims at anticipating the user's actions, and the second one is targeted at increasing the flexibility of the system whenever the user signals that there is an error he wants the system to correct.

Resumen

La principal meta de esta tesis es desarrollar sistemas de traducción asistida y de traducción automática que presenten mayor sinergia con sus usuarios potenciales. Por ello, el objetivo es hacer los sistemas estado del arte más ergonómicos, intuitivos y eficientes, con el fin de que el experto humano se sienta más cómodo al utilizarlos. Con este fin se presentan diferentes técnicas enfocadas a mejorar la adaptabilidad y el tiempo de respuesta de los sistemas de traducción automática subyacentes, así como también se presenta una estrategia cuya finalidad es mejorar la interacción hombre-máquina en un entorno de traducción interactiva. Todo ello con el propósito último de rellenar el hueco existente entre el estado del arte en traducción automática y las herramientas que los traductores humanos tienen a su disposición.

En lo que respecta al tiempo de respuesta de los sistemas de traducción automática, en esta tesis se presenta una técnica de poda de los parámetros de los modelos de traducción actuales, cuya intuición está basada en el concepto de segmentación bilingüe, pero que termina por evolucionar hacia una estrategia de re-estimación de dichos parámetros. Utilizando esta estrategia se obtienen resultados experimentales que demuestran que es posible podar la tabla de segmentos hasta en un 97%, sin mermar por ello la calidad de las traducciones obtenidas. Además, estos resultados son coherentes en diferentes pares de lenguas, lo cual evidencia que la técnica que se presenta aquí es efectiva en un entorno de traducción automática tradicional, y por lo tanto podría ser utilizada directamente en un escenario de post-edición. Sin embargo, los experimentos llevados a cabo en traducción interactiva son ligeramente menos convincentes, pues implican la necesidad de llegar a un compromiso entre el tiempo de respuesta y la calidad de los sufijos producidos.

Por otra parte, se presentan dos técnicas de adaptación, con el propósito de mejorar la adaptabilidad de los sistemas de traducción automática. La primera de ellas se centra en mejorar la adaptabilidad del modelo de lenguaje, mediante su subdivisión en varios modelos de lenguaje más pequeños, pero más específicos. Una vez hecho esto, tales submodelos se interpolan en tiempo de traducción en función de la frase de entrada en cuestión. Los

submodelos específicos son construidos o bien teniendo en cuenta información procedente de etiquetas supervisadas existentes en diferentes conjuntos de datos bilingües, o bien mediante estrategias de agrupamiento no supervisadas, con el propósito de descubrir determinados temas o estilos lingüísticos. La segunda técnica de adaptación que se presenta en esta tesis consiste en aplicar la adaptación predictiva Bayesiana a los modelos de traducción subyacentes en los sistemas de traducción automática actuales. A pesar de que los experimentos de adaptación se han llevado a cabo en un entorno de traducción automática pura, los resultados obtenidos son lo suficientemente prometedores como para implementar las técnicas desarrolladas en esta tesis en un entorno interactivo en el futuro cercano. Sin embargo, vale la pena recalcar que las técnicas presentadas aquí pueden ser implementadas tal cual en un escenario de traducción asistida, en el cual un sistema de traducción automática proporciona las traducciones que el usuario debe corregir y validar.

Por último, también se dedica una especial atención a mejorar la sinergia entre el experto humano y el sistema de traducción interactiva. Para ello, se estudian dos formas diferentes de realimentación débil, con la intención de mejorar la productividad del traductor humano. Con este fin, se presentan dos modificaciones al esquema tradicional de interacción. La primera pretende anticipar las acciones del usuario, mientras que la segunda tiene por finalidad mejorar la flexibilidad del sistema en el caso en que el usuario señale que hay un error que quiere que el sistema corrija.

Resum

El principal objectiu d'aquesta tesi és desenvolupar sistemes de traducció assistida i de traducció automàtica que presenten una major sinergia amb els seus usuaris potencials. Per tant, el propòsit és dissenyar sistemes més ergonòmics, intuïtius i eficients, amb la intenció de que l'expert humà es senti més còmode a l'hora d'emprar-los. Per arribar a aquest fi es presenten diferents tècniques enfocades a millorar l'adaptabilitat i el temps de resposta dels sistemes de traducció automàtica subjacents, així com també es presenta una estratègia per a millorar la interacció home-màquina en un entorn de traducció interactiva. Tot això amb el proposit últim d'emplenar el buit existent entre l'estat de l'art en traducció automàtica i les eines que tenen els traductors humans a la seva disposició.

Pel que fa al temps de resposta dels sistemes de traducció automàtica, en aquesta tesi es presenta una tècnica de poda dels paràmetres dels models de traducció actuals, la intuïció de la qual està basada en el concepte de segmentació bilingüe, però que acaba per evolucionar cap a una estratègia de re-estimació d'aquestos paràmetres. Emprant aquesta estratègia s'obtenen resultats experimentals que demostren que és possible podar la taula de segments fins un 97%, sense minvar amb això la qualitat de les traduccions obtingudes. A més, aquests resultats són coherents en diferents parells de llengües, la qual cosa evidencia que la tècnica que es presenta ací és efectiva en un entorn de traducció automàtica tradicional, i per tant podria ser utilitzada directament en un escenari de post-edició. No obstant això, els experiments duts a terme en traducció interactiva són lleugerament menys convincents, donat que impliquen la necessitat d'arribar a un compromís entre el temps de resposta i la qualitat dels sufixos produïts.

D'altra banda, es presenten dues tècniques d'adaptació, amb el propòsit de millorar l'adaptabilitat dels sistemes de traducció automàtica. La primera d'elles es centra en millorar l'adaptabilitat del model de llenguatge, mitjançant la seva subdivisió en diversos models de llenguatge més petits, però més específics. Una vegada fet això, eixos submodels s'interpolen en temps de traducció en funció de la frase d'entrada en qüestió. Els submodels específics

són construïts bé tenint en compte informació procedent d'etiquetes supervisades existents en diferents conjunts de dades bilingües, o bé mitjançant estratègies d'agrupament no supervisades, amb el propòsit de descobrir determinats temes o estils lingüístics. La segona tècnica d'adaptació que es presenta en aquesta tesi consisteix a aplicar l'adaptació predictiva Bayesiana als models de traducció subjacents als sistemes de traducció automàtica actuals. Tot i que els experiments d'adaptació s'han dut a terme en un entorn de traducció automàtica pura, els resultats obtinguts són prou prometedors com per implementar les tècniques desenvolupades en aquesta tesi en un entorn interactiu en el futur proper. Tot i això, val la pena recalcar que les tècniques desenvolupades en aquesta tesi poden ser implementades sense modificacions en un entorn de traducció assistida en el qual un sistema de traducció automàtica estadístic proporciona les traduccions que l'usuari haurà de modificar i validar.

Finalment, també es dedica especial atenció a millorar la sinergia entre l'expert humà i el sistema de traducció interactiva. Per a això, s'estudien dues formes diferents de realimentació feble, amb la intenció de millorar la productivitat del traductor humà. Amb aquesta finalitat, es presenten dues modificacions a l'esquema tradicional d'interacció. La primera pretén anticipar les accions de l'usuari, mentre que la segona té per finalitat millorar la flexibilitat del sistema en el cas en què l'usuari assenyali que hi ha un error i vol que el sistema corregeixi.

Preface

Machine translation is a thriving research field that has been receiving an increasing amount of attention with the up-rise of globalisation. Information technologies, and the popularisation of user-generated content such as assistance forums, have led big corporations to introduce the use of machine translation, with the purpose of making language-specific content available to all their potential customers, which are often located in different parts of the world and may not be able to understand one common language. However, machine translation is not only needed in fields where the amount of data is overwhelming, but also in fields where the bilingual data is perhaps less abundant, but translation quality is critical, such as foreign affairs, medicine or in the military domain. Hence, the need for more task-oriented machine translation systems arises. In these scenarios, it is often the case that machine translation systems need to collaborate closely with human experts, with the purpose of achieving high quality translations efficiently, giving rise to the popularisation of the computer assisted translation (CAT) and interactive machine translation (IMT) paradigms. In these scenarios, the interaction between the machine translation system and a human translator is crucial for obtaining high quality translations in an efficient manner. While CAT is a very broad research field covering all imaginable tools which can be made available to the human expert for lightening his job, IMT is a specific sub-field of computer-aided translation. Under this translation paradigm, the computer software that assists the human translator attempts to predict the text the user is going to input by taking into account all the information it has available. Whenever such prediction is wrong and the user provides feedback to the system, a new prediction is performed considering the new information available. Such process is repeated until the translation provided matches the user's expectations. This thesis explores three main problems that arise when attempting to build task-specific systems which are thought to be used within a computer assisted translation scenario: system performance, adaptability and usability.

In the first place, state-of-the-art statistical machine translation (SMT) systems are often

unable to yield real-time performance. This problem is even worse when the system has been trained on very large amounts of data, which is always desirable given that more data usually implies higher model coverage. When the amount of translation options and bilingual data made available to the system increases, translation throughput is necessarily affected, and model pruning strategies need to be applied with the purpose of not having the human translator waiting too long for the system to produce its output, which would be on the one hand exasperating, and on the other hand economically inefficient. In this thesis, we focused on proposing a model pruning strategy which proves to be able to decrease system response time drastically, while keeping translation quality within state-of-the-art ranges.

Another topic tackled in this thesis is system adaptability. There is extensive work in SMT which proves that the translation quality produced by a typical machine translation system drops significantly when the text to be translated stems from a different topic than the data which has been used to train the system. In addition, different human translators may have different styles when translating a document, which implies that lexical choice or sentence length may be required to vary even when working within one single domain. Furthermore, from a user point of view it is mentally exhausting for a human translator to correct the same mistakes over and over again, while having the impression that those same mistakes will keep on appearing because the system is not learning from its own errors. For these reasons, system adaptability is unveiled as a key feature within a machine translation system that is setup within a human-machine collaborative framework. In the present thesis, two different model adaptation techniques are presented. The first one deals with the problem of language model adaptation, i.e., adapting the specific part of the translation system that controls word ordering and structure in the hypotheses produced. The second one deals with the adaptation of the translation model itself, which is the part of the translation system that will account for lexical choice and sentence length, among other features. Although the techniques proposed in the current thesis are only applied in a classical machine translation scenario, they are perfectly suitable for usage within a computer assisted translation scenario, whenever the translation proposed to the user is generated by means of a typical statistical machine translation system. Applying the most promissing techniques developed within an interactive machine translation scenario is left for future work.

Lastly, usability of interactive machine translation systems is also a very important topic when attempting to build systems that are to be used by human users, whose expertise when using computers should not always be assumed. Hence, it is important to take special care when designing the interaction scheme, so that the human translator feels as comfortable as possible when using the translation interface. In this context, it is important to realise that the keyboard is not the only input device that the human user may use, but rather that richer interaction schemes might boost productivity. Nevertheless, it is also important to keep the interaction interface simple, so that the human expert is not overburdened. In this thesis, we propose a very simple and intuitive extension to the classical interactive machine translation interaction scheme, which takes into account the actions that the user performs before correcting any word of the proposed hypothesis.

The objective of this thesis is, hence, to confront three of the main problems that prevent IMT systems from being more widely used. More precisely, the scientific contributions of this thesis can be divided into three groups as follows:

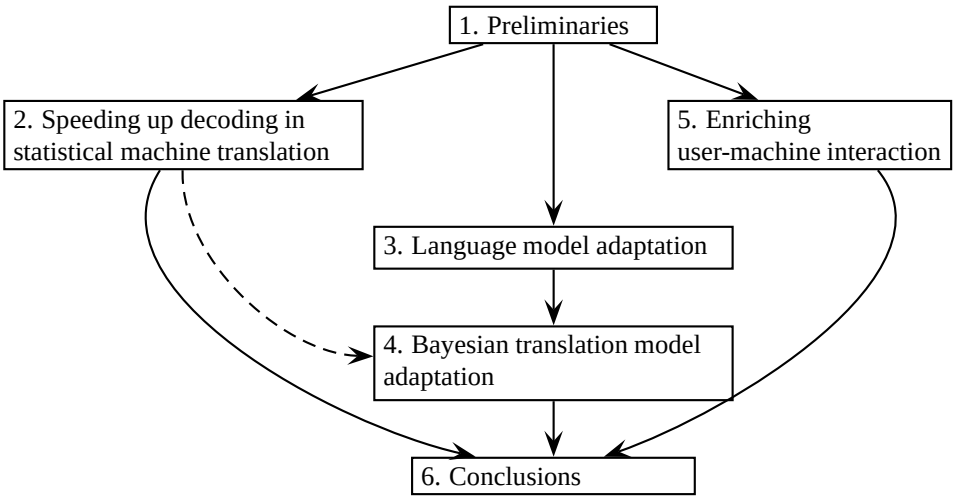
1. **Speeding up decoding in statistical machine translation.** A novel parameter pruning technique is presented. Such technique relies on the concept of bilingual segmentation for obtaining one single segmentation of each bilingual sentence present in the training corpus. This technique is then refined and re-oriented as a full parameter re-estimation strategy, which has as side-effect an important reduction of the computational resources required at translation time. Experimental results are reported on several different language pairs and involving both a SMT and an IMT framework.

2. **Language model adaptation.** Starting from the idea of bilingual clustering, we propose a novel method for performing language model adaptation within SMT. For doing this, the training data is first divided into different subsets. This subdivision step is either performed in a fully unsupervised manner, or by taking into account supervised labels present in different bilingual corpora. Assuming that each one of these subsets presents specific characteristics, such as topic or language style, specific sub-models are built from them. These smaller language models are then dynamically interpolated in translation time according to the text to be translated. Experiments are conducted in a classical SMT setting, involving several different language pairs and corpora.

3. **Bayesian translation model adaptation.** Bayesian predictive adaptation (BPA) is an adaptation strategy which has proved to be successful in different research areas where adaptation is needed. In this thesis, BPA is revised and its core ideas are applied within a classical SMT framework. For doing this, the theoretical formulation is first presented, for both a batch and an online adaptation scenario. Exhaustive experiments analysing BPA performance on different corpora are presented.

4. **Enriching user-machine interaction.** We study the possibility of considering the mouse as an additional interaction device between the machine translation back-end and the human user. Two different scenarios are considered: a first scenario in which the user does not need to be explicitly collaborative, and which takes advantage of the different actions performed by the user, and a second scenario in which a collaborative user is assumed, and which provides more flexibility to the final user interface. Experimental results within a simulated IMT environment are shown, involving different language pairs, for both extensions presented.

The above contributions are sequentially organised in 7 chapters that cover most of the work developed in this thesis. A sequential reading of the document is recommended if the readers wish to learn about the complete work. However, in case the readers be only interested in a specific research topic, they can also opt to read only the chapters that are related to that topic, taking into account the following dependency graph among chapters:



The parameter pruning strategy is proposed in Chapter 2. The two different approaches to this strategy are presented together with experimental results assessing the quality of the translations produced by the pruned systems.

The language model adaptation technique is presented in Chapter 3, in both its unsupervised and supervised forms. Then, the application of Bayesian predictive adaptation for translation model adaptation is presented in Chapter 4. Specifically, BPA is applied both in an online and in a batch adaptation setting, and for adapting either the log-linear model weights present in state-of-the-art SMT systems or the feature functions that are leveraged by such weights. In doing so, the fundamental equation of SMT is revised, so as to include the adaptation data and marginalise the model parameters.

The user-machine interaction scheme is revised in Chapter 5. Here, both modifications to the classical interaction scheme are presented, alongside with experimental results within a simulated IMT environment.

The final chapter, Chapter 6, summarises the conclusions that can be drawn from all the work described here, together with the work that still lies ahead and the most important scientific publications that have been derived from this thesis.

Contents

Abstract **vii**

Resumen **ix**

Resum **xi**

Preface **xiii**

Notation **xvii**

Contents **xix**

1 Preliminaries **1**
1.1 Introduction 3
1.2 Statistical machine translation 4
1.2.1 Phrase-based statistical machine translation 7
1.2.2 Statistical machine translation evaluation metrics 14
1.3 Interactive machine translation 17
1.3.1 IMT using word graphs 19
1.3.2 IMT evaluation metrics 20
1.4 Main bilingual corpora 21
1.5 Toolkits 23
Bibliography 25

2 Speeding up decoding in statistical machine translation **31**
2.1 Introduction 33
2.2 Related work 34

2.3 Bilingual segmentation . . . 34
2.4 True bilingual segmentation . . . 36
2.5 Source-driven bilingual segmentation . . . 38
2.6 Phrase-table pruning and parameter re-estimation . . . 39
2.7 Experimental results . . . 40
2.7.1 Bilingual segmentation experiments . . . 40
2.7.2 Parameter re-estimation experiments . . . 42
2.7.3 Interactive machine translation results . . . 47
2.8 Conclusions and future work . . . 49
Bibliography . . . 51

3 Language model adaptation for statistical machine translation 53
3.1 Introduction . . . 55
3.2 Related work . . . 55
3.3 General framework for language model adaptation . . . 56
3.4 Supervised labelled data for language model adaptation . . . 58
3.5 Unsupervised clustering for language model adaptation . . . 62
3.6 Weight optimisation strategies . . . 64
3.7 Experimental results . . . 65
3.7.1 Experiments using supervised labels . . . 66
3.7.2 Unsupervised clustering experiments . . . 74
3.8 Conclusions and future work . . . 78
Bibliography . . . 80

4 Bayesian translation model adaptation 83
4.1 Introduction . . . 85
4.2 Related work . . . 86
4.3 Bayesian predictive adaptation for SMT . . . 88
4.3.1 Adaptation of log-linear weights . . . 91
4.3.2 Adaptation of log-linear features . . . 92
4.4 Online Bayesian adaptation . . . 94
4.5 Bayesian adaptation for model stabilisation . . . 95
4.6 Sampling methods . . . 96
4.6.1 Heuristic sampling . . . 96
4.6.2 Gaussian sampling . . . 97
4.6.3 Markov chain Monte Carlo . . . 98
4.6.4 Viterbi-like approach . . . 100
4.7 Practical approximations . . . 100
4.8 Experiments . . . 102
4.8.1 Corpora . . . 102
4.8.2 Machine translation evaluation measures . . . 103
4.8.3 Batch adaptation results . . . 103
4.8.4 Online adaptation results . . . 116
4.8.5 Bayesian adaptation for system stabilisation . . . 118
4.9 Conclusions and future work . . . 119

Bibliography 123

5 Enriching user-machine interaction in IMT **127**
5.1 Introduction 129
5.2 Related work 129
5.3 Anticipated proposal as a form of weaker feedback 130
5.4 Partial refusal pointer action 131
5.5 Experimental results 133
5.6 Conclusions and future work 135
Bibliography 137

6 Conclusions **139**
6.1 Summary 141
6.2 Scientific publications 142
6.3 Future work 145
Bibliography 147

List of Figures **149**

List of Tables **155**

CHAPTER 1

Preliminaries

No man was ever wise by chance.

Lucio Anneo Seneca

Contents

1.1 Introduction **3**
1.2 Statistical machine translation **4**
1.3 Interactive machine translation **17**
1.4 Main bilingual corpora **21**
1.5 Toolkits **23**
Bibliography **25**

Todo lo que usted quiera, sí señor, pero son las palabras las que cantan, las que suben y bajan... Me prosterno ante ellas... Las amo, las adhiero, las persigo, las muerdo, las derrito... Amo tanto las palabras... Las inesperadas... Las que glotonamente se esperan, se escuchan, hasta que de pronto caen... Vocablos amados... Brillan como piedras de colores, saltan como platinados peces, son espuma, hilo, metal, rocío... Persigo algunas palabras... Son tan hermosas que las quiero poner todas en mi poema... Las agarro al vuelo, cuando van zumbando, y las atrapo, las limpio, las pelo, me preparo frente al plato, las siento cristalinas, vibrantes, ebúrneas, vegetales, aceitosas, como frutas, como algas, como ágatas, como aceitunas... Y entonces las revuelvo, las agito, me las bebo, me las zampo, las trituro, las emperejilo, las liberto... Las dejo como estalactitas en mi poema, como pedacitos de madera bruñida, como carbón, como restos de naufragio, regalos de la ola... Todo está en la palabra...

Confieso que he vivido. Pablo Neruda.

Everything you want, yessir, but it is the words that sing, the rise and fall... I prostrate before them... I love them, sticks them, the chase, bite them, the melt... I so love the words... The unexpected... Those who greedily hoped for, we hear, until suddenly fall... Fold loved... Sparkle like colored stones, platinum leap like fish, are foam, thread, metal, spray... I chase a few words... They are so beautiful that I put all my poem... The grip on the fly when they humming, and caught, clean the hair, I prepare against the plate, I feel clear, vibrant, Eburne, vegetables, oily, like fruit, like algae, like agates, like olives... And then stir, agitations, I did drink, I did zampa, crush, dress up, the freedom... I leave them in my poem like stalactites, like bits of polished wood, and coal, as a wreck, gifts of the wave... Everything is in the word...

I confess that I have lived. Google Translate.

1.1 Introduction

Natural language processing (NLP) is a research field of artificial intelligence and linguistics that is gaining more and more importance with the up-rise of computerised communication technologies. The mass usage of Internet and also the cheap storage possibilities that computers offer have given humanity the opportunity to record and store unprecedented amounts of linguistic data. At this moment, scientists estimate that the total amount of stored data is somewhere in the whereabouts of 295 exabytes (i.e. $295 \cdot 10^{18}$ bytes, or $295 \cdot 10^{6}$ terabytes). Moreover, the pace at which such data is growing keeps increasing. In order to cope with such a huge amount of data, computerised approaches dealing with it have become necessary. Of course, not all this data is susceptible to be processed by NLP systems. However, this symbolises the fact that, as the amount of data increases, NLP is elucidated as the only way in which such large amounts of data can be analysed.

Machine translation (MT) is a specific sub-field of NLP, and studies the way in which automatic systems should be developed so that they are able to translate a certain sentence in a source language into a sentence in a given target language, such that source and target sentences preserve the exact same meaning, while being both well-formed sentences in their respective languages. The idea of developing an automatic procedure by means of which a source text could be translated into a target language without the intervention of a human can be traced back to the 17th century, when René Descartes proposed a universal language which would be able to represent all ideas contained within any existing language. Since then, the idea of an *interlingua* to and from which the translation process is simple has been present in the MT community, although such a language has never been found.

More recently, after World War II and at the beginning of the Cold War, the Georgetown-IBM experiment achieved during January 1954 to gain a large amount of interest, both from the general public and from funding agencies, leading to the famous publication by Weaver (Weaver, 1955). Although the experiment was perceived as a success and the authors claimed that, with the appropriate funding, MT would be a well-solved problem within three or five years, the fact was that the experiment implied a system containing only six grammar rules and 250 vocabulary entries. As progress on MT evolved at a much slower pace than expected, funding was severely cut after the 1960 report of the ALPAC (Automatic Language Processing Advisory Committee) (Bar-Hillel, 1960).

The 1960 ALPAC report lead to drastic direction shift in MT research that led to the up-rise of rule-based machine translation (RBMT) (Hutchins, 1986) systems in the early 1970s. Such systems, which are currently loosing weight in the state of the art, rely on linguistic information of both source and target languages, which is basically retrieved from bilingual dictionaries and grammars. Two different RBMT paradigms were developed: transfer RBMT systems, which attempt to map the source language into the target language directly, and interlingual RBMT systems, which make use of an intermediate language which is assumed to be easy to translate into and from. Although RBMT systems are still in use, many of the commercial systems implementing RBMT are shifting towards statistical MT, such as Systran and Google translate.

It was not until the late 1980s that statistical machine translation (SMT), the pattern recognition approach to MT, transformed the state of the art in MT completely, by developing statistical models which were able to learn to translate between different languages

in a word-to-word fashion. It was then when the researchers at the IBM Thomas J. Watson Research Center contributed most significantly to the research in SMT by developing word-based statistical translation models (Brown et al., 1993), popularly known as IBM models, which are even nowadays used in current state-of-the-art SMT systems. Together with the introduction of phrase-based models (Koehn et al., 2003; Tomas and Casacuberta, 2001; Zens et al., 2002), word-alignment models were critical for the up-rise of SMT, which is nowadays the most dominant technology in MT.

In recent MT evaluations (Callison-Burch et al., 2011; Paul et al., 2010), the most dominant technology was the statistical approach to MT, which is the one that is currently receiving the most attention. Nevertheless, recent user reports (Hollowood, 2011; Yuste et al., 2010) claim that it is possible to achieve better results, from a user point of view, by combining both SMT and RBMT. This idea has recently given rise to the so-called hybrid MT technologies, which attempt to leverage the strengths of both paradigms.

1.2 Statistical machine translation

Statistical machine translation (SMT), systems have proved in the last years to be an important alternative to rule-based MT systems, being even able of outperforming commercial machine translation systems in the tasks they have been trained on (Callison-Burch et al., 2007). Moreover, the development effort behind a rule-based machine translation system and an SMT system is dramatically different, the latter being able to adapt to new language pairs with little or no human effort, whenever suitable corpora are available (Hutchings and Somers, 1992).

The grounds of modern SMT were established in (Brown et al., 1993), where the problem of machine translation was defined as the problem of translating a certain sentence $\boldsymbol{x}$ from a given source language into a target sentence $\boldsymbol{y}$, being

$$\begin{aligned} \boldsymbol{x} &= x_1 \dots x_j \dots x_J \qquad x_j \in \mathcal{X} \\ \boldsymbol{y} &= y_1 \dots y_i \dots y_I \qquad y_i \in \mathcal{Y} \end{aligned}$$

where x_j and y_i denote source and target words, each one belonging respectively to the source and target vocabularies, $\mathcal{X}$ and $\mathcal{Y}$. $J = |\boldsymbol{x}|$ and $I = |\boldsymbol{y}|$ are the lengths of the source and target sentences, respectively.

In SMT, it is assumed that every source string (or sentence) $\boldsymbol{x}$ may be the translation of every target string $\boldsymbol{y}$. Then, the key idea of SMT is to establish a procedure by means of which every pair of strings $(\boldsymbol{x}, \boldsymbol{y})$ is assigned a score $p(\boldsymbol{y}|\boldsymbol{x})$, which is interpreted as the probability that $\boldsymbol{y}$ is an appropriate translation for a given $\boldsymbol{x}$. Such procedure is the SMT model, which we will denote by $\mathcal{M}$, and then the probability of $\boldsymbol{y}$ being a translation of $\boldsymbol{x}$ is given by the expression

$$\Pr(\boldsymbol{y}|\boldsymbol{x}) \approx p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{M}) \tag{1.1}$$

$$= \frac{p(\boldsymbol{y}; \mathcal{M}) p(\boldsymbol{x} \mid \boldsymbol{y}; \mathcal{M})}{p(\boldsymbol{x}; \mathcal{M})} \tag{1.2}$$

where Bayes' theorem has been applied between Equation 1.1 and Equation 1.2. In the following, $\mathcal{M}$ will be assumed implicit, with the purpose of simplifying notation.

Once the SMT model has been established, translating a certain sentence $\boldsymbol{x}$ can be formulated as the problem of finding that specific sentence $\hat{\boldsymbol{y}}$ that maximises the probability given in Equation 1.2, i.e.

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, p(\boldsymbol{y} \mid \boldsymbol{x}) \tag{1.3}$$

$$= \underset{\boldsymbol{y}}{\operatorname{argmax}}\, p(\boldsymbol{x} \mid \boldsymbol{y}) \cdot p(\boldsymbol{y}) \tag{1.4}$$

where $p(\boldsymbol{x})$ does not affect the maximisation and has been hence neglected in Equation 1.4, which is often referred to as the fundamental equation of SMT, and also source-channel approach. Here, $p(\boldsymbol{y} \mid \boldsymbol{x})$ has been decomposed into two different probabilities: the *statistical language model* of the target language $p(\boldsymbol{y})$ and *the (inverse) translation model* $p(\boldsymbol{x} \mid \boldsymbol{y})$. Although it might seem odd to model the probability of the source sentence given the target sentence, this decomposition has a very intuitive interpretation: the translation model $p(\mathbf{x} \mid \mathbf{y})$ will capture the word or phrase relations between both input and output language, whereas the language model $p(\mathbf{y})$ will penalise ill-formed sentences of the target language.

The first translation models were word-based, i.e. source words were translated into one or more target words, and these words were then re-ordered so as to compose the final output sentence. For building this word-to-word correspondences, word alignments were introduced (Brown et al., 1993). In the inverse version of the word alignment models, a source word x_j is aligned to a set of target word positions $\boldsymbol{a}_j = \{i_1, \ldots, i_l\}$. From a generative perspective, such an alignment implies that source word x_j generates target words $y_{i_1}, \ldots, y_{i_l}$. Modelling the translation process in such a way requires using a hidden variable $\boldsymbol{a}$, since alignments cannot be observed in the training process, yielding:

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \sum_{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x}, \boldsymbol{y})} p(\boldsymbol{x}, \boldsymbol{a} \mid \boldsymbol{y}) \tag{1.5}$$

where $\mathcal{A}$ denotes the set of all possible alignments between $\boldsymbol{x}$ and $\boldsymbol{y}$.

A large number of different word-alignment models have been proposed. To start with, Brown et al. already proposed five different models in their seminal work in (Brown et al., 1993), with an increasing degree of complexity and which were intended to be trained sequentially by means of the Expectation-Maximisation (EM) (Dempster et al., 1977; Wu, 1983), each of them yielding good initial values for the next model. Hence, these five models were intended to be trained sequentially. In addition, other authors (Och, 2003; Vogel et al., 1996) proposed further models, which have also gained popularity. Figure 1.1 illustrates a typical alignment between an input and an output sentence.

However, an important breakthrough in SMT was achieved when the source-channel approach was replaced by a maximum entropy (Berger et al., 1996) modelling of the translation process. By modelling $p(\boldsymbol{y} \mid \boldsymbol{x})$ directly, it became possible to introduce a set of M different feature functions $h_m(\boldsymbol{x}, \boldsymbol{y})$ into the translation process (Och and Ney, 2002; Papineni et al., 1998), with $m = 1, \ldots, M$. Each feature function is then assigned a feature weight λ_m, which represents how important is feature h_m for the translation of $\boldsymbol{x}$ into $\boldsymbol{y}$. This approach

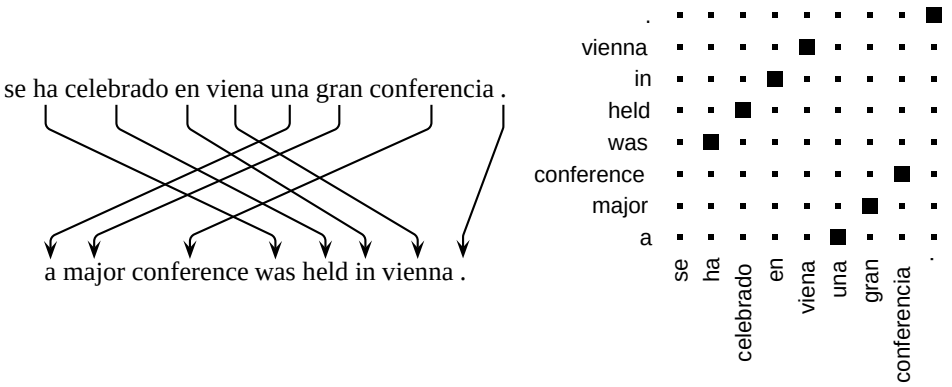


Figure 1.1: Example of word alignments computed automatically by means of a word-alignment model. The left side shows the alignment as a function of the source sentence (up) and the target sentence (down). On the right side, the alignment is shown in a matricial form, as is often done in SMT.

leads to the so-called log-linear models, where

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{\exp \sum_{m=1}^{M} \lambda_m h_m(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_{m=1}^{M} \lambda_m h_m(\boldsymbol{x}, \boldsymbol{y}')} \tag{1.6}$$

$$= \frac{\exp \boldsymbol{\lambda} \cdot \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \boldsymbol{\lambda} \cdot \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}')} \tag{1.7}$$

$h_m(\boldsymbol{x}, \boldsymbol{y})$ is a score function representing an important feature for the translation of $\boldsymbol{x}$ into $\boldsymbol{y}$, as for example the language model of the target language, a reordering model or several translation models. Typically, the largest part of the translation models included into current state-of-the-art SMT systems can be defined locally, but other models, such as reordering or language models, can only be defined at the sentence level due to longer-range dependencies among translation units. The weights $\boldsymbol{\lambda} = [\lambda_1 \dots \lambda_M]^T$ are normally optimised with the use of a development set.

In the expression above, it should be noted that the normalisation term present in Equation 1.6 has been omitted, since it is considered constant in the maximisation and is hence not needed when searching for the best output sentence $\hat{\boldsymbol{y}}$. This is important, since computing the normalisation term would be very costly. Nevertheless, such term is very often needed when developing more sophisticated approaches, as will be seen in Chapter 4.

Note that, in Equation 1.6, the feature functions $h_m(\boldsymbol{x}, \boldsymbol{y})$ are typically defined in the logarithmic domain. This means that, in the case that a certain feature represents a probability, the feature itself will be the logarithm of such probability. For example, if the feature is ought to represent the direct translation probability $p(\boldsymbol{y} \mid \boldsymbol{x})$, the feature itself will be $\log p(\boldsymbol{y} \mid \boldsymbol{x})$. Nevertheless, in practise some of the features actually do represent logarithms of probabilities, but others, as will be described in next section, do not.

Given Equation 1.6, the decision rule is given by the expression

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) \tag{1.8}$$

The use of log-linear models implied an important break-through in SMT, allowing for a significant increase in the quality of the translations produced. However, it should also be noted that the log-linear approach is actually a generalisation of the source-channel approach described above: if the set of features is limited to

$$\begin{aligned} h_1(\boldsymbol{x}, \boldsymbol{y}) &= \log p(\boldsymbol{y}) \\ h_2(\boldsymbol{x}, \boldsymbol{y}) &= \log p(\boldsymbol{x} \mid \boldsymbol{y}) \end{aligned}$$

and the corresponding weights are set to one, i.e., $\lambda_1 = \lambda_2 = 1$, searching for the optimum translation $\hat{\boldsymbol{y}}$ in Equation 1.8 is exactly equivalent to searching for $\hat{\boldsymbol{y}}$ in Equation 1.4.

1.2.1 Phrase-based statistical machine translation

One of the most popular instantiations of log-linear models in SMT are phrase-based (PB) models (Koehn, 2010; Koehn et al., 2003; Tomas and Casacuberta, 2001; Zens et al., 2002). PB models allow to capture contextual information to learn translations for whole phrases instead of single words. The basic idea of phrase-based translation is to segment the source sentence into phrases, then to translate each source phrase into a target phrase, and finally to re-order the translated target phrases in order to compose the target sentence. For this purpose, phrase-tables are produced, in which a source phrase is listed together with several target phrases and the probability of translating the former into the latter. PB models constitute nowadays the core of the state of the art in SMT, although more recent approaches, such as hierarchical models (Chiang, 2005) or finite state models (Casacuberta and Vidal, 2004) are able to yield similar translation quality (Callison-Burch et al., 2010; Koehn and Monz, 2006; Paul et al., 2010).

The model

The derivation of PB models stems from the concept of bilingual segmentation, i.e. sequences of source words and sequences of target words. Usually, it is assumed that only segments of contiguous words are considered, and that no overlap between such segments may exist. In such case the number of source segments is equal to the number of target segments (say K) and each source segment is aligned with only one target segment and vice versa.

From a generative point of view, the process of translating a source sentence into a target sentence by means of a PB SMT model is accomplished by means of the following steps:

1. Segment source sentence $\boldsymbol{x}$ into K source phrases $\{\tilde{x}_1 \dots \tilde{x}_k \dots \tilde{x}_K\}$.
2. Translate each one of the source phrases into target phrases $\{\tilde{y}_1 \dots \tilde{y}_k \dots \tilde{y}_K\}$.
3. Re-order the target phrases so as to build the final output sentence $\hat{\boldsymbol{y}}$.

Typically, some of the features included into PB-models can be defined at the local phrase level, such as the direct translation probability $p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{k=1}^{K} p(\tilde{y}_k \mid \tilde{x}_k)$. However, other features, such as the language model or phrase-reordering models cannot be defined at the translation unit level. We shall denote by $h^l(\cdot,\cdot)$ a feature which can be defined at the local phrase level, and, conversely, $h^s(\cdot,\cdot)$ will denote a feature which cannot be defined at the local phrase level. Let $\boldsymbol{\kappa}$ be a certain segmentation of sentence pair $(\boldsymbol{x},\boldsymbol{y})$, which segments such sentence pair into K phrases, such that

$$\begin{aligned} \boldsymbol{x} &= \tilde{x}_1 \dots \tilde{x}_k \dots \tilde{x}_K \qquad \tilde{x}_k \in \mathcal{B}(\boldsymbol{x}) \\ \boldsymbol{y} &= \tilde{y}_1 \dots \tilde{y}_k \dots \tilde{y}_K \qquad \tilde{y}_k \in \mathcal{B}(\boldsymbol{y}) \end{aligned}$$

where $\mathcal{B}(\boldsymbol{x}) \subseteq \mathcal{X}^+$ is the set of all possible sequences of contiguous words within sentence $\boldsymbol{x}$. Equivalently, $\mathcal{B}(\boldsymbol{y}) \subseteq \mathcal{Y}^+$ is the set of all possible segments (i.e. sequences of contiguous words) of the target sentence. Note that, by formulating PB models as above, the model is restricted to have the same amount of segments in both source and target sides of the bilingual sentence. This implies that source phrases must produce exactly one phrase in the target sentence. In addition, since $\mathcal{B}(\boldsymbol{x})$ and $\mathcal{B}(\boldsymbol{y})$ have been defined as a subset of the positive closure over alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively, empty phrases are not allowed, i.e. each phrase must contain at least one word. Although these two conditions are quite restrictive, these are a very common assumption made in order to make the search problem more tractable.

Then, the probability of sentence pair $(\boldsymbol{x},\boldsymbol{y})$ can be formulated as follows, separating and re-grouping those feature functions which can be defined at the local phrase level:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{\boldsymbol{\kappa}} p(\boldsymbol{\kappa}) \cdot p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\kappa}) \tag{1.9}$$

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{x};\boldsymbol{\kappa}) &= \frac{\exp \sum_{m=1}^{M} \lambda_m h_m(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_{m=1}^{M} \lambda_m h_m(\boldsymbol{x},\boldsymbol{y}')} \qquad (1.10) \\ &= \frac{\exp\{\sum_{m=1}^{M^l} \sum_{k=1}^{K} \lambda_m^l h_m^l(\tilde{x}_k,\tilde{y}_k) + \sum_{m=1}^{M^s} \lambda_m^s h_m^s(\boldsymbol{x},\boldsymbol{y})\}}{\sum_{\boldsymbol{y}'} \exp\{\sum_{m=1}^{M^l} \sum_{k=1}^{K} \lambda_m^l h_m^l(\tilde{x},\tilde{y}'_k) + \sum_{m=1}^{M^s} \lambda_m^s h_m^s(\boldsymbol{x},\boldsymbol{y}')\}} \\ &= \frac{\exp\{\sum_{k=1}^{K} \sum_{m=1}^{M^l} \lambda_m^l h_m^l(\tilde{x}_k,\tilde{y}_k) + \sum_{m=1}^{M^s} \lambda_m^s h_m^s(\boldsymbol{x},\boldsymbol{y})\}}{\sum_{\boldsymbol{y}'} \exp\{\sum_{k=1}^{K} \sum_{m=1}^{M^l} \lambda_m^l h_m^l(\tilde{x}_k,\tilde{y}'_k) + \sum_{m=1}^{M^s} \lambda_m^s h_m^s(\boldsymbol{x},\boldsymbol{y}')\}} \\ &= \frac{\exp\{\sum_{k=1}^{K} g^l(\tilde{x}_k,\tilde{y}_k) + g^s(\boldsymbol{x},\boldsymbol{y})\}}{\sum_{\boldsymbol{y}'} \exp\{\sum_{k=1}^{K} g^l(\tilde{x}_k,\tilde{y}'_k) + g^s(\boldsymbol{x},\boldsymbol{y})\}} \qquad (1.11) \end{aligned}$$

In this last expression, $g^l(\cdot,\cdot)$ represents the combination of features defined at the local phrase level, each one weighted accordingly, and $g^s(\cdot,\cdot)$ represents the combination of features which cannot be defined at the local phrase level.

Although Equation 1.9 implies that all possible segmentations of the candidate hypothesis need to be computed upon search, in practise the Viterbi segmentation is used, and only the maximum probability segmentation is taken into consideration. If the probability of the segmentation $p(\boldsymbol{\kappa})$ is considered constant, such approximations lead to the following decision

rule:

$$\hat{\boldsymbol{y}} \quad = \quad \underset{\boldsymbol{y},\boldsymbol{\kappa}}{\operatorname{argmax}} \sum_{k=1}^{K} g^l(\tilde{x}_k, \tilde{y}_k) + g^s(\boldsymbol{x}, \boldsymbol{y}) \tag{1.12}$$

where the normalisation denominator in Equation 1.11 has been neglected because it has no influence on the maximisation. Although it is not common in SMT literature to separate local and global features, this will be useful later on in other chapters of this thesis.

Learning phrase-based models

The most important step when learning a PB model is to compute a *phrase-table*, which is a translation table containing each one of the phrase pairs $(\tilde{x}, \tilde{y})$ observed during training, alongside with the value of each one of the local feature functions.

Hence, the first step when learning PB models is to extract phrase pairs from a sentence-aligned bilingual corpus. In the last years, a wide variety of heuristic techniques to produce PB models have been researched and implemented (Koehn et al., 2003). Firstly, a direct learning of the inverse translation model $p(\boldsymbol{x}|\boldsymbol{y})$ was attempted (Marcu and Wong, 2002; Tomas and Casacuberta, 2001). Other approaches have suggested exploring more linguistically motivated techniques (Sánchez and Benedí, 2006; Watanabe et al., 2003). However, the one technique which has been more widely adopted involves the heuristic extraction of phrase pairs (Zens et al., 2002), in which all phrase pairs coherent with a given word alignment are extracted. In most cases, one of the IBM alignments described in previous section is used for this purpose. Since these word alignments are very restrictive because each target word is assigned only zero or one source words, source-to-target and target-to-source alignments are combined heuristically. This procedure is often called *symmetrisation*. Once this is done, the set of phrases consistent with the symmetrised word alignments is extracted from every sentence pair in the training set. An illustration of how this is done can be seen in Figure 1.2

Most typically, the different local features $h_m(\cdot, \cdot)$ that are included into the translation table are:

- Inverse translation probability, given by the formula

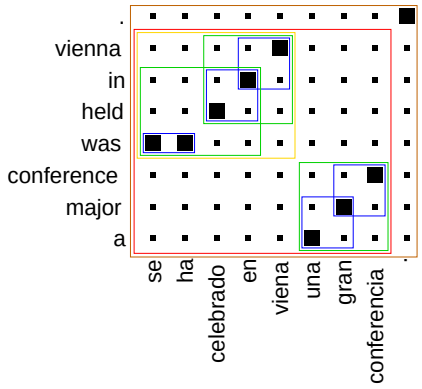
$$p(\tilde{x} \mid \tilde{y}) = \frac{C(\tilde{x}, \tilde{y})}{C(\tilde{x})} \tag{1.13}$$

  where $C(\tilde{x}, \tilde{y})$ is the number of times segments $\tilde{x}$ and $\tilde{y}$ were extracted throughout the whole corpus, and $C(\tilde{x})$ is the count for phrase $\tilde{x}$.

- Direct translation probability, $p(\tilde{x} \mid \tilde{y})$, which is obtained analogously.

- Inverse and direct lexicalised features, $w(\tilde{x} \mid \tilde{y})$, which attempt to account for the lexical soundness of each phrase pair, estimating how well each of the words in one language translates to each of the words in the other language. These lexicalised features were defined in (Zens et al., 2002)

- A constant feature, or *phrase penalty*, whose purpose is to avoid the use of many small phrases in decoding time, and favour the use of longer ones. Typically, this feature is set to number e.



se ha	was
celebrado	held
en	in
viena	vienna
una	a
gran	major
conferencia	conference
.	.
celebrado en	held in
en viena	in vienna
una gran	a major
gran conferencia	major conference
se ha celebrado en	was held in
celebrado en viena	held in vienna
una gran conferencia	a major conference
⋮	⋮

Figure 1.2: Example of how consistent phrases are extracted from a word alignment. On the left, the alignment matrix after symmetrisation is shown. Black squares represent word alignments, whereas extracted phrases are marked with a rectangle involving one or more squares. On the right, the phrases that would be extracted from that matrix. Note that word *se* cannot be extracted on its own because its alignment requires word *ha* to be extracted together with it so as to preserve alignment consistency.

In addition to the local features, typical state-of-the-art SMT systems also include a re-ordering model. In fact, the non-monotonicity problem in translation is one of the toughest problems that SMT systems need to face. Different languages entail different word order, and systems which do not tackle the re-ordering problem in any way are mostly unable to yield satisfactory results when translating between language pairs from different origin. This problem has been well-known in SMT for some time, and (Berger et al., 1996) already introduced in their alignment models what they called distortion models, in an effort towards including in their SMT system a solution for the re-ordering problem. (Vilar et al., 1996), tried to partially solve the problem by monotonising the most probable non-monotone alignment patterns and adding a mark in order to be able to remember the original word order. (Kumar and Byrne, 2005) learnt weighted finite state transducers accounting for local re-orderings of two or three positions. Other works, such as (Kanthak et al., 2005; Zens et al., 2004), dealt with input sentence re-ordering, where the main idea is to reorder the input sentence in such a way that the translation model will not need to account for possible word re-orderings. Other works (Xiong et al., 2006) deal with the re-ordering problem from a maximum entropy point of view, establishing a re-ordering model based on a set of features which the authors consider to be important for assessing the (non-) monotonicity of two specific phrases.

However, the re-ordering model which has found perhaps the most widespread acceptance among PB SMT systems is the one proposed in (Koehn et al., 2005), where a lexicalised re-ordering model is proposed. Let $(\tilde{x}_k, \tilde{y}_k)$ be the current phrase being consid-

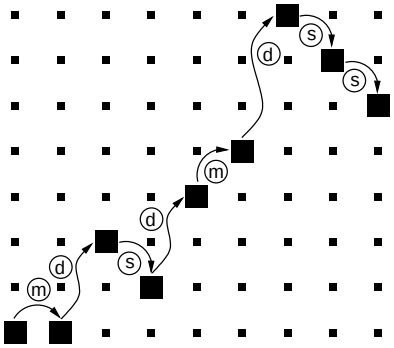


Figure 1.3: Alignment matrix with the different re-ordering types. *m* stands for monotone, *s* stands for swap, and *d* stands for discontinuous.

ered, $(\tilde{x}_{k-1}, \tilde{y}_{k-1})$ the previous one, in the order established by the source sentence, and $(\tilde{x}_{k+1}, \tilde{y}_{k+1})$ the next phrase pair to be translated by the decoding algorithm. Three possible re-ordering types, also called orientations, are considered: monotone, swap, and discontinuous. A swap occurs when inverting the order between phrase $(\tilde{x}_k, \tilde{y}_k)$ and the previous phrase $(\tilde{x}_{k-1}, \tilde{y}_{k-1})$ would result in a monotonic ordering of the phrases, and a discontinuity whenever such swap would still yield a non-monotonic ordering. Figure 1.3 shows examples of these three classes of orientations. Then, the probability of a given phrase pair $(\tilde{x}_k, \tilde{y}_k)$ having a certain orientation o with respect to the previous phrase $(\tilde{x}_{k-1}, \tilde{y}_{k-1})$ is given, following the maximum likelihood principle, by

$$p(o \mid \tilde{x}_k, \tilde{y}_k) = \frac{C(o, \tilde{x}_k, \tilde{y}_k)}{\sum_{o'} C(o', \tilde{x}_k, \tilde{y}_k)} \tag{1.14}$$

where $C(o, \tilde{x}_k, \tilde{y}_k)$ is the number of times that phrase pair $(\tilde{x}_k, \tilde{y}_k)$ has been observed to appear in orientation o with respect to the previous phrase in the training data.

In addition to considering the orientation with respect to phrase pair $(\tilde{x}_{k-1}, \tilde{y}_{k-1})$, it is also common to include into the model the probability of phrase pair $(\tilde{x}, \tilde{y})$ presenting a certain orientation with respect to $(\tilde{x}_{k+1}, \tilde{y}_{k+1})$. Since this implies the estimation of a large amount of parameters, it may lead to sparsity issues. For this reason, $p(o \mid \tilde{x}_k, \tilde{y}_k)$ is typically smoothed by the prior of orientation o.

The re-ordering model is an example of feature which cannot be defined at the local phrase level, since it depends on the position of the phrase translated before the current phrase. Other non-local features also include the language model and a word penalty, which attempts to regulate the fertility of the source words.

To sum up, typical features present in most state-of-the-art PB SMT systems include fourteen different feature functions h_i:

- the five local features described above, i.e. $p(\tilde{x} \mid \tilde{y})$, $p(\tilde{y} \mid \tilde{x})$, $w(\tilde{x} \mid \tilde{y})$, $w(\tilde{y} \mid \tilde{x})$ and number e
- the six probabilities defined by the lexicalised re-ordering model when considering the orientation with the previous and with the next phrase. In addition, it is also common to

include an exponential function penalising very long-range re-orderings. This accounts to a total of seven feature functions belonging to the re-ordering model.

- the language model
- the word penalty

Tuning in phrase-based models

Once the bilingual phrases have been extracted from a sentence aligned bilingual corpus, the features $\boldsymbol{h}$ described in the previous section can already be computed. However, at this point it is still necessary to obtain an appropriate value for the scaling factors $\boldsymbol{\lambda}$. The process of obtaining such a vector is often called *tuning*. To this end, numerous methods have been proposed. For instance, (Watanabe et al., 2007) propose to use of the margin infused relaxed algorithm (MIRA) (Crammer et al., 2006) for the specific task of adjusting $\boldsymbol{\lambda}$. More recently, (Sokolov and Yvon, 2011) proposed to view the tuning problem as a set of operations over a specific semi-ring. Alternatively, (Hopkins and May, 2011) proposed to view the problem as a ranking problem, where each step of the tuning procedure consists in deciding whether a given translation hypothesis should be ranked lower or higher within the set of possible hypotheses that are provided by the search procedure. Similarly, (Martínez-Gómez et al., 2011) propose to view the problem as a regression problem, where the problem of tuning is re-defined as a regression problem in which the log-linear combination in Equation 1.6 should approximately fit the translation quality function used.

However, perhaps the most popular approach for adjusting the scaling factors is the one proposed in (Och, 2003), commonly referred to as minimum error rate training (MERT). This algorithm implements a coordinate-wise global optimisation and consists on two basic steps. First, n-best hypotheses are extracted for each one of the sentences of a given development set. Next, the optimum $\boldsymbol{\lambda}$ is computed so that the best hypotheses in the n-best list, according to a reference translation and a given metric, are the ones that the search algorithm would produce. Since it is often the case that there is not a single $\boldsymbol{\lambda}$ vector that would promote all the best hypothesis throughout the whole development set to the first position in the n-best list, a compromise is often achieved, in which the specified metric is maximised. These two steps are iteratively repeated until convergence, where $\boldsymbol{\lambda}$ remains unchanged.

Decoding in phrase-based models

Once the model for PB translation has been established according to Equation 1.11 and the appropriate decision rule has been stated in Equation 1.12, an algorithm is needed for carrying out the maximisation described and establishing which is the best candidate hypothesis $\boldsymbol{y}^*$ that should be produced as final translation. However, the search problem in SMT has been shown to be an NP-complete problem (Knight, 1999; Udupa and Maji, 2006), which implies that different approximations and simplifications need to be made in order to deal with the problem efficiently. To this end, different algorithmic solutions have been proposed, such as the multi-stack depth-first decoding algorithm (Ortiz-Martínez, 2011) proposed by (Berger et al., 1996) for word-based models, greedy strategies (Germann et al., 2001), or dynamic programming solutions (García-Varea, 2003).

However, the decoding algorithm which has found the most widespread acceptance in SMT is the one proposed by (Tillmann and Ney, 2003), and which is an adaptation to SMT of the classic beam search algorithm proposed in (Jelinek, 1998) for speech recognition. In this algorithm, the translation is generated sequentially from left to right, and re-ordering between source and target phrases happens when the next source phrase to be translated, $\tilde{x}_k$, is not located directly after the one that has just been translated, $\tilde{x}_{k-1}$. A typical procedure for translating a certain input sentence is exemplified in Figure 1.4. In this figure, the initial (empty) hypothesis is first expanded into several partial hypotheses by using the different phrases extracted in Figure 1.2. The use of these phrases leads to different coverage vectors, denoted in the figure by $\kappa_{\boldsymbol{x}}$, indicating which words of the source sentence have already been translated. The reason for keeping track of which words have already been translated is double: on the one hand, for the purpose of not accounting for a given source word twice in the translation hypothesis; on the other hand, because in this kind of algorithm only hypotheses with the same amount of source words covered compete with each other. Given that the probability of a certain hypothesis is computed as a product, the more the amount of source words translated, the less the probability mass assigned to that specific hypothesis. Since hypothesis expansion is done by expanding first those hypotheses with the most probability, the algorithm would keep expanding hypotheses with few translated words. This is conveniently solved by means of the coverage vector by allowing to compete among each other only those hypotheses with the same amount of translated words. For example, in Figure 1.4, the hypotheses that would compete among each other would be ③ and ⑤. Note that it is not normal to have the same sentence both for training and for test: although such a thing could eventually happen, in this case the same sentence is used for illustrative purposes.

Coverage problems in phrase-based SMT

As described in Section 1.2.1, phrase extraction is typically done by a heuristic procedure, which attempts to extract a rather large amount of phrases from the bilingual sentences seen in training. However, given that the heuristic algorithm employed relies on word-alignments and on the concept coherent phrases, it might be possible that phrases which actually do appear in the training data, but are not considered coherent may end up resulting as unseen for the SMT system. This means, in practise, that the SMT system trained may actually be unable to account for the correct output sentence $\boldsymbol{y}^{\tau}$. Furthermore, given the large number of segments that are extracted from each bilingual sentence, the maximum word length of a phrase is often restricted for performance reasons, and following common knowledge that establishes that longer phrases tend to never be seen again.

If the training data was composed only by the bilingual sentence in Figure 1.2, a word as simple as the Spanish word *se* (a reflexive pronoun) would be considered out of vocabulary by the PB SMT system, even though such word was actually seen in training. More dramatic is the example shown in Figure 1.5. In this example, which has been extracted from a real training procedure, only three phrase pairs will be extracted, and the remaining words will not be included into the PT. The problem here can be easily exemplified by looking at the word *cannot*, which presents multiple alignments. In order to include target word *cannot* within a consistent alignment, one would need to include word *puedo* into the alignment, but including word *puedo* implies that word *I* is also included. Including *I* also forces the two

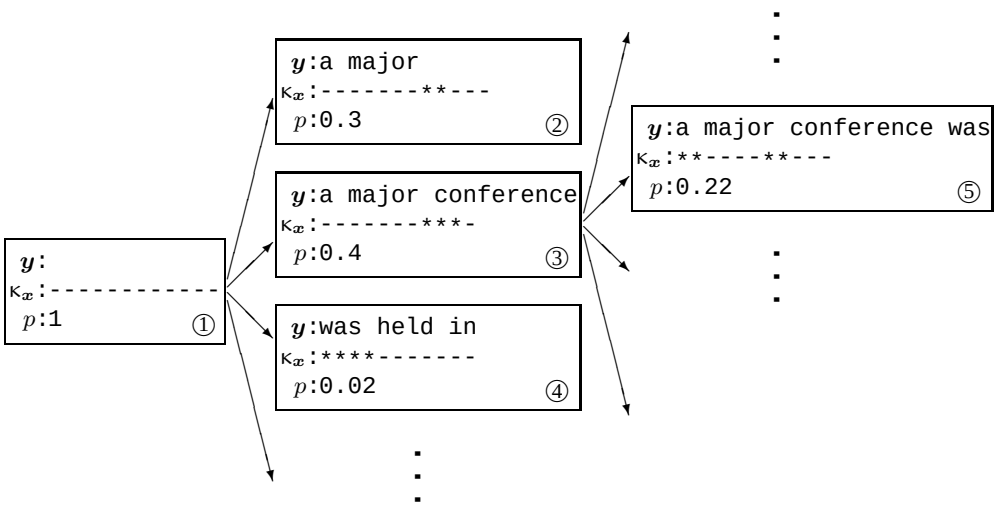


Figure 1.4: Example of decoding procedure following the phrases extracted in Figure 1.2, with the input sentence being "`Se ha celebrado en viena una gran conferencia .`". $\kappa_{\boldsymbol{x}}$ illustrates the coverage vector of that specific partial hypothesis. The coverage vector $\kappa_{\boldsymbol{x}}$ of a specific hypothesis keeps track of which words of the source sentence $\boldsymbol{x}$ have been translated until that point, so that words that already have their counterpart in the target sentence $\boldsymbol{y}$ are not translated again. In this figure, character `-` at the n-th position specifies that source word x_n has not been translated yet, and `*` indicates that already has. The probability p of each hypothesis is only for illustrative purposes.

commas to be included, together with whatever words appear between both. Continuing with this procedure leads to the necessity of including the whole sentence pair (except for the final dot) as a phrase before being able to include *cannot* into a consistent alignment. However, as explained above, it is quite common to restrict the maximum length of the phrases to be extracted. If such maximum is set to e.g. 7, the complete sentence pair will not be included into the system, and *cannot* will remain unknown despite having been observed in training.

As will be seen in the other chapters, the coverage problem will be an issue when dealing with the different techniques and algorithms described in this thesis, and different approximations will be needed to confront it. So as to provide a coarse idea about the importance of the coverage problem, this problem implies that a state-of-the-art SMT system is not able to produce the reference present in a bilingual corpus in about 30% to 80% of the cases, depending on the specific corpus being considered.

1.2.2 Statistical machine translation evaluation metrics

Evaluation in SMT is a very controversial issue. On the one hand, human evaluation is way too costly for experimentation purposes. Having a human translator assess the quality of the output produced by a SMT system for every combination of parameters that need to be adjusted in tuning time would render research in SMT unfeasible. This leads to the

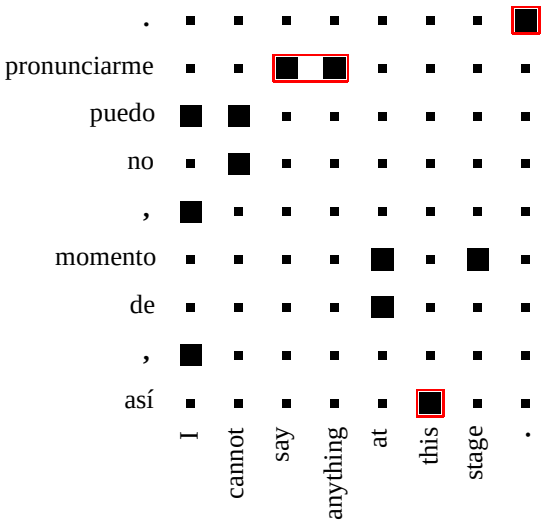


Figure 1.5: Example of word alignment that results in coverage problems. Maximum phrase length of 7 is assumed. Black squares represent word alignments, whereas extracted phrases are marked with a rectangle involving one or more squares.

wide-spread use of automatic evaluation metrics that are very cheap to use. On the other hand, however, there is a growing feeling in the MT community that claims that current SMT systems are not optimising translation quality as such, but are rather optimising a given evaluation metric without taking into consideration the real impact on the usability of the translations produced. This is due mainly to the problem of not having a reference sentence which can be considered ground truth, as is the case in other NLP research fields such as speech recognition or handwritten character recognition. This implies that it is often very difficult to assess how good a certain SMT output is, even for humans.

Many different evaluation metrics have been proposed, and this issue has even been the topic of recent SMT workshops (Callison-Burch et al., 2010, 2011). Typically, the main goal when designing automatic SMT evaluation metrics is to achieve a metric presenting a high correlation with human judgements of translation quality. However, even this is often questioned, specially when taking into account that inter-annotator agreement is often low (Callison-Burch et al., 2011).

In this thesis, SMT output will be evaluated by means of BLEU (Papineni et al., 2001) and TER (Snover et al., 2006), which are two of the most popular evaluation metrics employed in SMT.

BLEU *(Bilingual Evaluation Understudy) score*: This score measures the precision of unigrams, bigrams, trigrams, and four-grams with respect to a set of reference translations, with a penalty for too short sentences (Papineni et al., 2001). BLEU is not an error rate, i.e. the higher the BLEU score, the better. BLEU can be single- or multi-reference, but in the present thesis only single-reference BLEU will be used due to corpus restrictions. In practise, BLEU implements a geometrical average of n-gram precision. The consequence of this is that BLEU is often only well-defined on the corpus level, but

not on the sentence level. Consider for instance a sentence of three words. Such sentence will never share a common four-gram with the reference sentence, and BLEU will score zero even when the hypothesis produced by the system and the reference sentence are identical. As will be seen in further chapters, this may lead to problems when attempting to identify the best translation for a single input sentence. BLEU will be reported as a percentage, ranging from 0 to 100.

TER *(Translation Edit Rate)*: Translation Error Rate (Snover et al., 2006) is an error metric for MT that measures the number of edits required to change a system output into one of the references. TER is computed as the minimum number of edits required to modify the system hypothesis so that it matches the reference translation, normalised by the average number of reference words. In this case, possible edits include insertion, deletion, substitution of single words and shifts of word sequences. In the original work, the authors claimed that single-reference TER correlates as well with human judgements of MT quality as the four-reference variant of BLEU. As in BLEU, TER can also be multi-reference, but in this thesis single-reference TER will be used. TER will also be reported as a percentage, although it can yield values over 100.

In addition to BLEU and TER results, confidence interval sizes will also be provided, with the purpose of assessing whether differences in BLEU and TER are statistically significant or not. To this end, the methods described in (Koehn, 2004) will be followed. Specifically, two different statistical significance tests will be used, both relying on bootstrap re-sampling.

- *Test-specific* bootstrap re-sampling. Typically, for establishing a confidence interval for a given score it would be necessary to translate a certain (large) number of different test sets. However, if only one test set $\mathcal{E}$ of size $|\mathcal{E}|$ is available, an equivalent approach consists in drawing from $\mathcal{E}$ a random sample of sentences of size $|\mathcal{E}|$, with repetition. After evaluating the translation quality of such sample, the procedure is repeated b times, where b depends on the precision we would like for the confidence interval. If a precision of one decimal digit is desired, then $b = 1000$, and if two decimal digits are requested, then $b = 10,000$. Once b random samplings are extracted, and their translation quality has been assessed, all b scores are sorted. Dropping the upper 2.5% of the scores obtained yields the upper bound for the 95% confidence interval, and dropping the lower 2.5% yields the lower bound for the 95% confidence interval. Then, under the assumption that the sentences within the test set $\mathcal{E}$ are independent, we have the certainty that the *true* score that the SMT system tested would obtain would be within that interval 95% of the times.

- *Paired* bootstrap re-sampling. The previous bootstrap re-sampling technique is appropriate for evaluating the confidence on the score provided by a certain system. However, if we are interested in establishing whether a certain SMT system A performs better than another system B, regardless of where the true score may lie, then we need to perform *paired* bootstrap re-sampling. This is done by sampling the test set at random, in the same way as described above, but this sampling is performed on both systems at the same time, i.e., the $|\mathcal{E}|$ sentences sampled will be translated by both systems A and B at the same time. Then, the difference in score, $\mu(\mathcal{E}, A) - \mu(\mathcal{E}, B)$, between

both systems will be measured, and it is such difference which will be sorted, and from which the confidence interval will be obtained. Hence, and again under the assumption that the sentences within $|\mathcal{E}|$ are independent, if both upper and lower bounds of the confidence interval are positive, it can be said that system A performs better than system B 95% of the times, and if both such bounds are negative, it can be said that B performs (significantly) better than A.

For reporting confidence intervals in this thesis, an efficient implementation of the two methods above was used. The key idea for performing such implementation relies in performing the bootstrap re-sampling on the sentence-level counts which lead to the translation scores used, and not on the sentences as such. Hence, much computational effort is saved, since it is not needed to translate b test sets, obtain such counts, and then compute the final translation quality scores; the only thing needed is to repeat the computation of the final scores b times. For this reason, obtaining the confidence intervals ends up being very cheap, and hence the confidence intervals reported in this thesis were obtained after performing $b = 10,000$ bootstrap re-sampling repetitions, unless stated otherwise.

1.3 Interactive machine translation

Information technology advances in modern society have led to the need of more efficient methods of translation. It is important to remark that current MT systems are not able to produce ready-to-use texts (Arnold, 2003; Hutchins, 1999; Kay, 1997). 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 computer-assisted translation (CAT) paradigm. CAT offers different approaches in order to benefit from the synergy between humans and MT systems.

An important contribution to interactive CAT technology was carried out around the TransType (TT) project (Foster, 2002; Foster et al., 2002; Langlais et al., 2002; Och, 2003). 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 was to embed data driven MT techniques within the interactive translation environment.

Following these TT ideas, (Barrachina et al., 2009; Ortiz-Martínez, 2011) propose the usage of fully-fledged statistical MT (SMT) systems 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 SMT system as additional information to achieve further, hopefully improved suggestions. In this thesis, we also focus on the interactive and predictive, statistical MT (IMT) approach to CAT. The IMT paradigm fits well within the *interactive pattern recognition* framework introduced in (Romero et al., 2011; Vidal et al., 2007).

Figure 1.6 illustrates a typical IMT session. Initially, the user is given an input sentence $\boldsymbol{x}$ to be translated. The reference $\boldsymbol{y}$ provided is the translation that the user would like to achieve at the end of the IMT session. At iteration 0, the user does not supply any correct

SOURCE $(\boldsymbol{x})$**:**		Para encender la impresora:
REFERENCE $(\boldsymbol{y})$**:**		To power on the printer:
ITER-0	$(\boldsymbol{p})$	( )
	$(\hat{\boldsymbol{s}}_h)$	*To switch on:*
ITER-1	$(\boldsymbol{p})$	To
	$(\boldsymbol{s}_l)$	*switch on:*
	(k)	power
	$(\boldsymbol{s}_h)$	*on the printer:*
ITER-2	$(\boldsymbol{p})$	To power on the printer:
	$(\boldsymbol{s}_l)$	( )
	(k)	(#)
	$(\hat{\boldsymbol{s}}_h)$	( )
FINAL	$(\boldsymbol{p} \equiv \boldsymbol{y})$	To power on the printer:

Figure 1.6: IMT session to translate a Spanish sentence into English. Non-validated hypotheses are displayed in italics, whereas accepted prefixes are printed in normal font.

text prefix $\boldsymbol{p}$ to the system, for this reason $\boldsymbol{p}$ is shown as empty. Therefore, the IMT system has to provide an initial complete translation $\boldsymbol{s}_h$, as if it were a conventional SMT system. At the next iteration, the user validates a prefix $\boldsymbol{p}$ as correct by positioning the cursor in a certain position of $\boldsymbol{s}_h$. In this case, after the word "*To*". Implicitly, he is also marking the rest of the sentence, the suffix $\boldsymbol{s}_l$, as potentially incorrect. Next, he introduces a new word k, which is assumed to be different from the first word s_{l_1} in the suffix $\boldsymbol{s}_l$ which was not validated, i.e., $k \neq s_{l_1}$. This being done, the system suggests a new suffix hypothesis $\hat{\boldsymbol{s}}_h$, subject to $\hat{s}_{h_1} = k$. Again, the user validates a new prefix, introduces a new word and so forth. The process continues until the whole sentence is correct, which is validated introducing the special word "#". In this example, a potential user of the IMT system would have typed only one word out of five. Assuming that, without the IMT system, the user would have had to translate the whole sentence, the potential benefit consists in an effort reduction of 80%. If a post-edition environment is assumed as baseline, the user would have typed three words, versus only one in the case of IMT, leading to an effort reduction of 66% with respect to post-edition.

As the reader could devise from the IMT session described above, IMT aims at reducing the effort and increasing the productivity of translators, while preserving high-quality translation. For instance, in Figure 1.6, only three interactions were necessary in order to achieve the reference translation.

Formally, IMT is specified as an evolution of the SMT framework, and hence its formulation stems from the so-called fundamental equation of SMT, i.e., Equation 1.3. However, this equation needs to be modified according to the IMT scenario in order to take into account the part of the target sentence that is already translated, that is $\boldsymbol{p}$ and k:

$$\hat{\boldsymbol{s}}_h = \underset{\boldsymbol{s}_h}{\operatorname{argmax}}\, Pr(\boldsymbol{s}_h | \boldsymbol{x}, \boldsymbol{p}, k) \tag{1.15}$$

where the maximisation problem is defined over the suffix $\boldsymbol{s}_h$. This allows us to rewrite

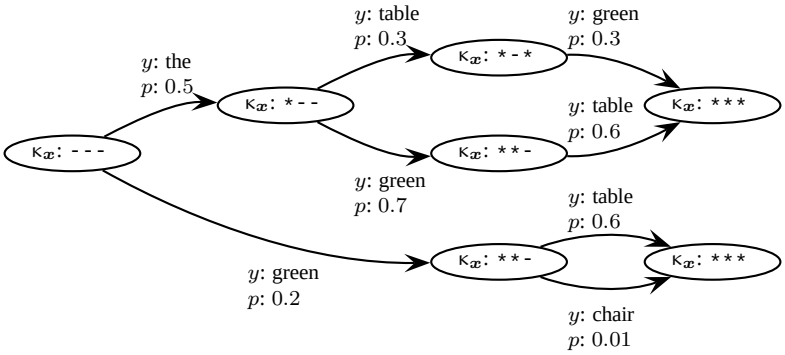


Figure 1.7: Example of word graph illustrating the translation of *la mesa verde*. $\kappa_{\boldsymbol{x}}$ is the coverage vector of the input sentence (see Section 1.2.1), where symbol - indicates an uncovered word, and symbol * an input word that has already been translated. Each edge is labelled with both the word emitted when transiting through that edge, and the probability assigned. Note that, for the sake of simplicity, this word graph is not a real example generated during a true search process.

Eq. 1.15, by decomposing the right side appropriately and eliminating constant terms, achieving the equivalent criterion

$$\hat{\boldsymbol{s}}_h = \underset{\boldsymbol{s}_h}{\operatorname{argmax}}\, Pr(\boldsymbol{p}, k, \boldsymbol{s}_h|\boldsymbol{x}). \tag{1.16}$$

An example of the intuition behind these variables is shown in Figure 1.6.

Note that, since $(\boldsymbol{p}\, k\, \boldsymbol{s}_h) = \boldsymbol{y}$, Eq. 1.16 is very similar to Eq. 1.3. The main difference is that the argmax search is now performed over the set of suffixes $\boldsymbol{s}_h$ that complete $(\boldsymbol{p}\, k)$, instead of complete sentences ($\boldsymbol{y}$ in Eq. 1.3). This implies that we can use the same models if the search procedures are adequately modified (Barrachina et al., 2009).

The phrase-based approach presented in Section 1.2.1 can be easily adapted for its use in an IMT scenario. The most important modification is to rely on a word graph that represents possible translations of the given source sentence. The use of word graphs in IMT has been studied in (Barrachina et al., 2009) in combination with two different translation techniques, namely, the alignment templates technique (Och and Ney, 2004; Och et al., 1999), and the Stochastic Finite State Transducers technique (Casacuberta and Vidal, 2007).

1.3.1 IMT using word graphs

Word graphs (Ueffing et al., 2002) have been successfully applied for a long time in other natural language processing fields, such as speech recognition (Ortmanns et al., 1997) and natural language generation (Knight and Hatzivassiloglou, 1995). A word graph is a weighted directed acyclic graph, composed out of nodes and edges. Each node represents one or more partial translation hypotheses (see Figure 1.4). In this case, we say one or more because different hypotheses may be grouped into a same node if they share the same coverage vector $\kappa_{\boldsymbol{x}}$ and the same completion options. Then, the edges connecting nodes represent transitions between such nodes, and are labelled each with one word of the target sentence, $\boldsymbol{y}_i$, and

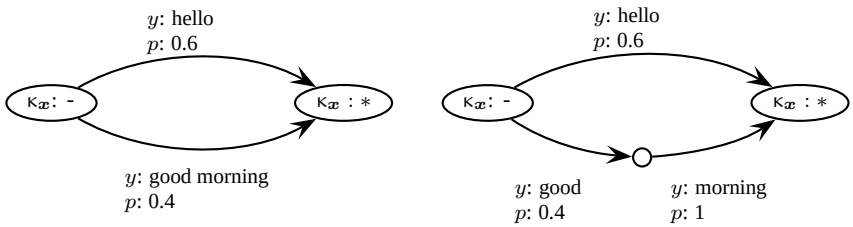


Figure 1.8: Example of conversion of a phrase graph (left) into a word graph (right).

weighted by a score assigned by the translation model, which evaluates how likely it is to emit word $\boldsymbol{y}_i$ after having already emitted the current partial translation hypothesis. In (Och, 2003), the use of a word graph is proposed as interface between an alignment-template SMT model and the IMT engine. Analogously, in this thesis, a word graph built during the search procedure performed on a PB SMT model will be used.

During the search process performed by the beam search algorithm (Section 1.2.1), it is possible to create a *phrase graph*. In such a graph, each node represents a state of the SMT model, and each edge a weighted transition between states labelled with a sequence of target words. Whenever a hypothesis is expanded, a new edge connecting the state of that hypothesis with the state of the extended hypothesis is added. The new edge is labelled with the sequence of target words that has been incorporated to the extended hypothesis and is weighted appropriately by means of the score given by the SMT model. Once the phrase graph is generated, it can be easily converted into a word graph by the introduction of artificial states for the words that compose the target phrases associated to the edges. Figure 1.8 illustrates an example of this procedure.

During the process of IMT for a given source sentence, the system makes use of the word graph generated for that sentence in order to complete the prefixes accepted by the human translator. Specifically, the system finds the best path in the word graph associated with a given prefix so that it is able to complete the target sentence, being capable of providing several completion suggestions for each prefix.

A common problem in IMT arises when the user sets a prefix which cannot be found in the word graph, since in such a situation the system is unable to find a path through the word graph and provide an appropriate suffix. The common procedure to face this problem is to perform a tolerant search in the word graph. This tolerant search uses the well known concept of Levenstein distance in order to obtain the most similar string for the given prefix (see (Ortiz-Martínez, 2011) for more details).

1.3.2 IMT evaluation metrics

As explained in Section 1.2.2, automatic evaluation of results is a difficult problem in MT. In fact, it has evolved to a research field with own identity. This is due to the fact that, given an input sentence, a large amount of correct *and* different output sentences may exist. Hence, there is no sentence which can be considered ground truth, as is the case in speech or text recognition. By extension, this problem is also applicable to IMT.

The two metrics most commonly used in IMT are:

WSR *Word Stroke Ratio*: This metric is computed as the quotient between the number of word-strokes a user would need to perform in order to achieve the translation he has in mind and the total number of words in the sentence (Barrachina et al., 2009). In this context, a word-stroke is interpreted as a single action, in which the user types a complete word, and is assumed to have constant cost. Moreover, each word-stroke also takes into account the cost incurred by the user when reading the new suffix provided by the system.

KSR *Key Stroke Ratio*: Similarly as for WSR, KSR measures the total number of key-strokes a user would need to perform before validating the final translation, divided by the total number of characters present in the sentence (Barrachina et al., 2009). KSR is clearly an optimistic measure, since in the scenario proposed the system is constantly proposing translation options after every key stroke, and the user is often overwhelmed by receiving a great amount of information. However, since the time taken by the user to read all those hypothesis is not considered, KSR may not be measuring the user's effort accurately. For these reasons, in the present thesis we favour the use of WSR, instead of KSR.

1.4 Main bilingual corpora

Given that SMT needs huge bilingual sentence-aligned corpora for training the statistical models that lie at the ground of the SMT system, this technology benefited greatly from the existence of multinational organisations, such as the Canadian Parliament, the European Parliament, or the United Nations, which need to translate the proceedings of their meetings into all the languages which are official within the core of such organisations. One of the first real-sized corpora that appeared was the Canadian Hansards corpus, which was the corpus used in the original works that established the fundamentals of SMT (Brown et al., 1993).

Since then, many corpora have been developed and gathered, some of them being smaller but more task-specific than the Canadian Hansards corpus, but other corpora preserving general domain have become very large, nourished by the increasing number of multinational organisations which translate their documentation into different languages.

The most important corpus used throughout the present work is the Europarl corpus (Koehn, 2005). This corpus is built from the transcription of European Parliament speeches published on the web. The data was collected in the 11 official languages of the European Union, in the period comprised between 1996 and 2010. It was obtained by crawling the web, then it was aligned at the document level and split into sentences, normalised, tokenised and aligned at the sentence level. This corpus has found a very widespread use in the SMT community, and has been used for numerous SMT evaluation campaigns (Callison-Burch et al., 2011; Paul et al., 2010). One main advantage of the Europarl corpus when compared with other similar-sized corpora is that the Europarl corpus can be downloaded for free. Given that this corpus increases in size year after year because of its nature, some of the experiments conducted during the time taken to elaborate this thesis were conducted on the second version of the corpus, while others were conducted on the third version, after such

		De	En	Es	En	Fr	En
WMT07 training	Sentences	751k		731k		688k	
	Running words	15.3M	16.1M	15.7M	15.2M	15.6M	13.8M
	Average length	20.3	21.4	21.5	20.8	22.7	20.1
	Vocabulary size	195k	66k	103k	64k	80k	62k
WMT10 training	Sentences	1219k		1272k		1251k	
	Running words	24.9M	26.1M	27.5M	26.6M	28.1M	25.6M
	Average length	20.4	21.4	21.6	20.9	22.5	20.5
	Vocabulary size	255k	82k	126k	83k	101k	81k
Devel.	Sentences	2000		2000		2000	
	Running words	55k	59k	61k	59k	67k	59k
	Average length	27.6	29.3	30.3	29.3	33.6	29.3
	OoV wrt WMT07	432	125	208	127	144	138
	OoV wrt WMT10	348	103	164	99	99	104
Devtest	Sentences	2000		2000		2000	
	Running words	54k	58k	60k	58k	66k	58k
	Average length	27.1	29.0	30.2	29.0	33.1	29.3
	OoV wrt WMT07	377	127	207	125	139	133
	OoV wrt WMT10	310	111	172	112	114	112
Test	Sentences	3064		3064		3064	
	Running words	82k	85k	92k	85k	101k	85k
	Average length	26.9	27.8	29.9	27.8	32.9	27.8
	OoV wrt WMT07	1020	488	470	502	536	519
	OoV wrt WMT10	825	404	383	419	424	415

Table 1.1: Characteristics of Europarl for each of the sub-corpora. OoV stands for "Out of Vocabulary" words with respect to (wrt) the specified training corpus. Devel. stands for Development, k for thousands of elements and M for millions of elements.

version was released, with the purpose of providing state-of-the-art quality results. In order to make the results reported in the present thesis comparable with other results reported in other works, standard partitions of the corpus will be used. Such partitions are the ones established in the 2007 Workshop on Statistical Machine Translation (WMT) (Callison-Burch et al., 2007) of the Association for Computational Linguistics in the case of the version 2 of the Europarl corpus, and the partition established for the 2010 WMT (Callison-Burch et al., 2010) in the case of version 3. Statistics for the language pairs used in the present work are provided in Table 1.1. The `Devtest` partition is the test set that was provided for the 2007 WMT for internal evaluation purposes, and `Test` partition is the set used for the final evaluation. At this point, it is important to point out that the `Test` partition included a *surprise* out-of-domain subset, which is the reason why the number of out-of-vocabulary words is so high for that specific set. The out-of-domain subset was extracted from the News-Commentary corpus (see next paragraph).

		De	En	Es	En	Fr	En
Training	Sentences	86.9k		80.9k		67.6k	
	Running words	1.8M	1.8M	1.8M	1.6M	1.6M	1.4M
	Average length	21.2	20.7	22.5	20.1	23.1	20.0
	Vocabulary size	86.7k	40.8k	53.5k	38.8k	43.3k	35.6k
NC 08	Sentences	2051		2051		2051	
	Running words	47.2k	49.8k	52.6k	49.8k	55.4k	49.8k
	Average length	23.0	24.3	25.7	24.3	27.0	24.3
	OoV wrt training	2941	1445	1781	1493	1736	1593
	OoV wrt WMT10	2015	962	1028	955	998	961
NC 09	Sentences	2525		2525		2525	
	Running words	62.7k	65.6k	68.1k	65.6k	72.6k	65.6k
	Average length	24.8	26.0	27.0	26.0	28.7	26.0
	OoV wrt training	3629	1853	2467	1916	2478	2035
	OoV wrt WMT10	2410	1247	1357	1229	1446	1247
NC 10	Sentences	2489		2489		2489	
	Running words	61.3k	61.9k	65.5k	61.9k	70.5k	61.9k
	Average length	24.6	24.9	26.3	24.9	28.3	24.9
	OoV wrt training	4056	1923	2404	2004	2312	2081
	OoV wrt WMT10	2834	1349	1394	1327	1375	1353

Table 1.2: Characteristics of the three News-Commentary test sets that will be used. `Training` refers to the News-Commentary training set. OoV stands for "Out of Vocabulary" words with respect to (wrt) the specified training corpus. NC stands for News-Commentary, k for thousands of elements and M for millions of elements.

Another corpus that will be used in several chapters is the News-Commentary corpus [a]. This corpus was obtained from different news feeds and was used as test set for the WMT in all its editions after year 2007. For this reason, results on different test sets will be reported, although standard partitions will always be respected. This corpus will be used mainly for test purposes, but the training partition of the corpus will also be used. Characteristics are provided in Table 1.2.

In addition, other smaller corpora will also be used for the purpose of evaluating the techniques described in some specific chapters. Given that these corpora will only be used in isolated occasions, their description will be given in that specific chapter.

1.5 Toolkits

For conducting the experiments reported in this thesis, several different NLP toolkits have been used, with the purpose of focusing on the main ideas which motivate this thesis. These toolkits are, mainly, the two SMT toolkits Moses and Thot, the word-alignment toolkit

[a] available from http://www.statmt.org/wmt11

GIZA++ and the language modelling toolkit SRILM.

Moses SMT toolkit

Moses (Koehn et al., 2007) is an open source SMT toolkit, licensed under the LGPL license, which includes a large amount of tools for training and optimisation of PB SMT systems, as well as a decoder for translating source texts by means of the models built. Recent versions of Moses also include a tree-based SMT system, although in this thesis only the PB SMT system will be used. The most standard setup provides all the feature functions described in Section 1.2.1, including the lexicalised re-ordering model described. Unless stated otherwise, this will be the standard setup used throughout this thesis for establishing the experimental baselines for assessing the techniques proposed.

Thot SMT toolkit

Thot (Ortiz-Martínez et al., 2005) is also a toolkit to train PB SMT models and is licensed under the GPL license. As GPL software, Thot only includes software to train SMT models. However, since Thot has been developed at the Universitat Politècnica de València, the present work benefited of internal versions which also include a decoder and a phrase aligner. In contrast with Moses, however, Thot does not include lexicalised re-ordering models, and re-ordering is limited to the an exponential function on the distance. Nevertheless, although lexicalised re-ordering models have evolved to become a standard when translating between European languages, the benefit in translation quality introduced is scarce, which means that results achieved by means of Thot are very near to the state of the art.

GIZA++ word-alignment toolkit

GIZA++ (Och and Ney, 2003) is a SMT toolkit that implements training and search for IBM models 1-5 and HMM. It also includes other tools which become handy when working in SMT, such as a tool to generate word classes or a tool to transform a corpus made out of strings into a numeric format. Since GIZA++ is used to build the word-alignments which are the key step when inferring a phrase-table, GIZA++ is used by both Moses and Thot.

SRI Language Modelling toolkit

SRILM (Stolcke, 2002) is a toolkit for building and applying statistical LMs, and is currently under development since 1995 by the Stanford Research Institute (SRI) Speech Technology and Research Laboratory. It also underwent important changes within the John Hopkins University/CLSP summer workshops in 1995, 1996, 1997, and 2002. Although SRILM includes a set of executable programs and scripts for performing the most standard tasks when modelling language, it also provides a wide range of libraries which can be used independently of the binaries.

Bibliography

Doug J. Arnold. *Why translation is difficult for computers*. John Benjamins Pub Co, 2003.

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

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, and Enrique Vidal. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28, 2009.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1): 39–71, 1996.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of machine translation. *Computational Linguistics*, 19(2): 263–311, 1993.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. (meta-) evaluation of machine translation. In *Proceedings of the ACL second workhop on Statistical Machine Translation*, pages 136–158, June 23 2007.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, and Omar Zaidan, editors. *Proceedings of the ACL joint fifth workshop on Statistical Machine Translation and Metrics MATR*. Association for Computational Linguistics, July 15–16 2010. URL `http://www.aclweb.org/anthology/W10-17`.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar F. Zaidan, editors. *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh, Scotland, July 2011. URL `http://www.aclweb.org/anthology/W11-21`.

Francisco Casacuberta and Enrique Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, 2004.

Francisco Casacuberta and Enrique Vidal. Learning finite-state models for machine translation. *Machine Learning*, 66(1):69–91, 2007.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 263–270, June 25–30 2005.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7 (Mar):551–585, 2006.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (1):1–38, 1977.

George Foster. *Text Prediction for Translators*. PhD thesis, Université de Montréal, 2002.

George Foster, Philippe Langlais, and Guy Lapalme. User-friendly text prediction for translators. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 148–155, July 6–7 2002.

Ismael García-Varea. *Traducción automática estadística: modelos de traducción basados en máxima entropía y algoritmos de*

búsqueda. PhD thesis, Universitat Politècnica de València, 2003.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. Fast decoding and optimal decoding for machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 228–235, July 6–11 2001.

Fred Hollowood. A tale of two technologies. In *Proceedings of the conference of the European Association for Machine Translation*, page 1, May 30–31 2011.

Mark Hopkins and Jonathan May. Tuning as ranking. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, July 27–31 2011.

W. John Hutchings and Harold L. Somers. *An Introduction to Machine Translation*. Academic Press, 1992.

John Hutchins. Retrospect and prospect in computer-based translation. In *Proceedings of the Machine Translation Summit X*, pages 30–44, September 13–17 1999.

W. John Hutchins. *Machine translation: past, present, future*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

Frederick Jelinek. *Statistical Methods for Speech Recognition (Language, Speech and Communication)*. MIT Press, 1998.

Stephan Kanthak, David Vilar, Evgeny Matusov, Richard Zens, and Hermann Ney. Novel reordering approaches in phrase-based statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 167–174, June 29–30 2005.

Martin Kay. It's still the proper place. *Machine Translation*, 12(1–2):35–38, 1997.

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

Kevin Knight and Vasileios Hatzivassiloglou. Two-level, many-paths generation. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 252–260, June 26–30 1995.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 388–395, July 25–26 2004.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Machine Translation Summit X*, pages 79–86, September 12–16 2005.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010. ISBN 978-0521874151.

Philipp Koehn and Christof Monz. Manual and automatic evaluation of machine translation between european languages. In *Proceedings of the NAACL Workshop on Statistical Machine Translation*, pages 102–121, June 8–9 2006.

Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, May 27 – June 1 2003.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-burch, Miles Osborne, and David Talbot. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *Proceedings of the international Workhop on Spo-*

ken Language Translation, October 24–25 2005.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics: Demo and Poster Sessions*, pages 177–180, June 23–30 2007.

Shankar Kumar and William Byrne. Local phrase reordering models for statistical machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 161–168, October 6–8 2005.

P.Philippe Langlais, Guy Lapalme, and Marie Loranger. Transtype: Development-evaluation cycles to boost translator's productivity. *Machine Translation*, 15(4):77–98, 2002.

Daniel Marcu and Daniel Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 133–139, July 6–7 2002.

Pascual Martínez-Gómez, German Sanchis-Trilles, and Francisco Casacuberta. Online learning via dynamic reranking for computer assisted translation. In *Proceedings of the international conference on Intelligent Text Processing and Computational Linguistics*, pages 93–105, February 20–26 2011.

Franz J. Och. Minimum error rate training for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 160–167, July 7–12 2003.

Franz J. Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 295–302, July 6–12 2002.

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

Franz J. Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(1):417–449, 2004.

Franz J. Och, Christof Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 20–28, June 21–22 1999.

Daniel Ortiz-Martínez. *Advances in Fully-Automatic and Interactive Phrase-Based Statistical Machine Translation*. PhD thesis, Universitat Politècnica de València, 2011.

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. Thot: a toolkit to train phrase-based statistical translation models. In *Proceedings of the Machine Translation Summit X*, pages 141–148, September 12–16 2005.

Stefan Ortmanns, Hermann Ney, and Xavier Aubert. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, 11(1): 43–72, 1997.

Kishore Papineni, Salim Roukos, and Todd Ward. Maximum likelihood and discriminative training of direct translation models. In *Proceedings of the international conference on Acoustics, Speech and Signal Processing*, pages 189–192, May 12–15 1998.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Technical Report RC22176 (W0109-022)*, 2001.

Michael Paul, Marcello Federico, and Sebastian Stüker. Overview of the IWSLT 2010 evaluation campaign. In *Proceedings of the international Workhop on Spoken Language Translation*, pages 3–27, December 2–3 2010.

Verónica Romero, Alejandro H. Toselli, and Enrique Vidal. *Multimodal Interactive Handwritten Text Transcription*. Series in Machine Perception and Artificial Intelligence (MPAI). World Scientific Publishing, 2011.

Joan A. Sánchez and Jose M. Benedí. Stochastic inversion transduction grammars for obtaining word phrases for phrase-based statistical machine translation. In *Proceedings of the NAACL Workshop on Statistical Machine Translation*, pages 130–133, June 8–9 2006.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of conference of the Association for Machine Translation in the Americas*, pages 223–231, August 8–12 2006.

Artem Sokolov and François Yvon. Minimum error rate training semiring. In *Proceedings of the conference of the European Association for Machine Translation*, pages 241–248, May 30–31 2011.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of the 7th international conference on Spoken Language Processing, 2002*, pages 901–904, September 16–20 2002.

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

Jesús Tomas and Francisco Casacuberta. Monotone statistical translation using word groups. In *Proceedings of the Machine Translation Summit X*, pages 357–361, September 18–22 2001.

Raghavendra Udupa and Hemanta K. Maji. Computational complexity of statistical machine translation. In *Proceedings of the conference of the European Chapter of the Association for Computational Linguistics*, pages 25–32, April 3–6 2006.

Nicola Ueffing, Franz J. Och, and Hermann Ney. Generation of word graphs in statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 156–163, July 6–7 2002.

Enrique Vidal, Luis Rodríguez, Francisco Casacuberta, and Ismael García-Varea. Interactive pattern recognition. In *Proceedings of the Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, pages 60–71, June 28–30 2007.

Juan M. Vilar, Enrique Vidal, and Juan C. Amengual. Learning extended finite-state models for language translation. In *Proceedings of Extended Finite State Models Workshop*, pages 92–96, August 11–12 1996.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. In *Proceedings of the international conference on Computational Linguistics*, volume 2, pages 836–841, August 5–9 1996.

Taro Watanabe, Eiichiro Sumita, and Hiroshi G. Okuno. Chunk-based statistical translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 303–310, July 7–12 2003.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. Online large-margin training for statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 764–773, June 28–30 2007.

Warren Weaver. *Translation*. MIT Press, Cambridge, MA., 1955.

C. F. Jeff Wu. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

Deyi Xiong, Qun Liu, and Shouxun Lin. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the international conference on Computational Linguistics and the annual meeting of the Association for Computational Linguistics*, pages 521–528, July 17–21 2006.

Elia Yuste, Manuel Herranz, Antonio Lagarda, Lionel Tarazón, Isaías Sánchez-Cortina, and Francisco Casacuberta. Pangeamt - putting open standards to work... well. In *Proceedings of conference of the Association for Machine Translation in the Americas*, October 31 – November 4 2010.

Richard Zens, Franz J. Och, and Hermann Ney. Phrase-based statistical machine translation. In *Proceedings of Advances in Artificial Intelligence: the annual German conference on Artificial Intelligence*, pages 18–32, September 16–20 2002.

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. Reordering constraints for phrase-based statistical machine translation. In *Proceedings of the international conference on Computational Linguistics*, pages 205–211, August 23–27 2004.

CHAPTER 2

Speeding up decoding in statistical machine translation

Pour examiner la vérité, il est besoin, une fois dans sa vie, de mettre toutes choses en doute autant qu'il se peut.

René Descartes

Contents

2.1 Introduction **33**
2.2 Related work **34**
2.3 Bilingual segmentation **34**
2.4 True bilingual segmentation **36**
2.5 Source-driven bilingual segmentation **38**
2.6 Phrase-table pruning and parameter re-estimation **39**
2.7 Experimental results **40**
2.8 Conclusions and future work **49**
Bibliography **51**

– Bonjour, dit le petit prince.

– Bonjour, dit le marchand.

C'était un marchand de pilules perfectionées qui apaisent la soif. On en avale une par semaine et l'on n'éprouve plus le besoin de boire.

– Pourquoi vends-tu ça? dit le petit prince.

– C'est un grosse économie de temps, dit le marchand. Les experts ont fait des calculs. On épargne cinquante-trois minutes par semaine.

– Et que fait-on de ces cinquante-trois minutes?

– On en fait ce que l'on veut...

« Moi, se dit le petit prince, si j'avais cinquante-trois minutes à dépenser, je marcherais tout doucement vers une fontaine... »

Le Petit Prince. Antoine de Saint-Exupéry.

– Hello, said the little prince.

– Hello, said the merchant.

He was a merchant for the ultimate pills that quench thirst. Are swallowed by a weeks and we feel no need to drink.

– Why are you selling? said the little prince.

– It's a big savings in time, said the merchant. The experts have calculated.

We save fifty-three minutes a week.

– And what about those fifty-three minutes?

– We do what we want...

"I, said the little prince, if I had fifty-three minutes to spend, I would walk slowly into a fountain... "

The Little Prince. Google Translate.

2.1 Introduction

Nowadays, the key step of the process of statistical machine translation (SMT) involves inferring a large table of phrase pairs that are translations of each other from a large corpus of aligned sentences. The set of all phrase pairs, together with estimates of conditional probabilities and other useful features, is included into the phrase-table. Such phrases are applied during the decoding process, combining their target sides to form the final translation.

A variety of algorithms to extract phrase pairs has been proposed (Marcu and Wong, 2002; Och and Ney, 2000, 2003; Ortiz-Martínez et al., 2008; Vogel, 2005; Zens et al., 2002). Typically, these algorithms heuristically collect a highly redundant set of phrases from each training sentence pair generating phrase-tables with a huge number of elements.

This bulk comes at a cost. Large phrase-tables lead to large data structures that require more resources and more time to process. More importantly, the large computational cost that such complex structures entail often implies that SMT systems are not able to yield real-time translation speed, which is crucial for the wide-spread implementation of PB IMT systems within modern CAT systems. Typical SMT systems will take several seconds to translate a certain input sentence, depending on the length of the sentence to be translated, but also on the amount of bilingual data made available at training time: the more training data, the larger the phrase-table that is estimated. In addition, effort spent in handling large tables could likely be more usefully employed in more features or more sophisticated search processes. Finally, this is also the main restriction for the widespread application of SMT techniques in small portable devices like cell phones, PDAs or hand-held game consoles; one can imagine many scenarios that could benefit from a lightweight translation device: tourism, medicine, military, etc.

In this chapter, it is shown that it is possible to prune the phrase-table by removing those phrase pairs that have little influence on the final translation performance. The present approach consists in selecting only those phrase pairs extracted from the most probable segmentation of the training sentences, which are the ones that are likely to be used during decoding time.

The technique presented here has several advantages. In the first place, it does not depend on the actual algorithm used to extract the phrase pairs, and therefore it can be applied to every imaginable method that assigns probabilities to phrase pairs. In addition, it provides a straightforward method for pruning the phrase-tables, without the need of adjusting any additional parameter. Moreover, it does not significantly affect translation quality, as measured by BLEU or TER scores, while very substantial savings in terms of computational requirements are reported.

The rest of the chapter is organised as follows. Section 2.2 reviews previously published techniques to prune the phrase-table. Section 2.3 reviews the bilingual segmentation problem in order to present our technique to filter the phrase-table. A solution taking into account both source and target sentence information is provided in Section 2.4. Then, a source-driven solution for that same problem is, in turn, provided in Section 2.5. This source-driven solution is revised in Section 2.6, by focusing more on the problem confronted, leading to a novel formula for the estimation of phrase-pairs within the phrase-table. Experiments are presented in Section 2.7, and the conclusions drawn from them are presented in Section 2.8. Future work yet to be done is also presented in this last section.

2.2 Related work

Most phrase-based decoders already include several built-in thresholds in order to prune the size of phrase-tables estimated from training corpora (Koehn et al., 2007; Ortiz-Martínez et al., 2005). They are usually related either to absolute scores of phrase pairs in the phrase-table or to relative scores between the phrase pairs sharing their source phrase.

Apart from phrase-table threshold pruning techniques, which are usually employed in SMT, different complementary methods in order to reduce even more the size of phrase-tables have been explored within the last years. For instance, (Johnson et al., 2007) propose to use significance testing in order to select only those phrase pairs which are the most co-occurring ones in the training corpus. In their experiments, they show that they are able to reduce the phrase-table in about 90% without any loss in translation quality. However, they also report that such percentage seems to decrease with larger corpora, since in larger corpora the amount of phrases with high frequency counts increases. In this chapter, we present a phrase-table pruning technique which is able to reduce the phrase-table in about 97%. Even though the experimental conditions are different, we consider the difference in reduction and in methodology to be significant. However, future work will involve a more close comparison between the technique presented in (Johnson et al., 2007) and the methods presented in the current chapter.

Another work approaching this problem, inspired by the optimal brain damage algorithm, relies on the idea of usage statistics. For this purpose, (Eck et al., 2007) suggest to translate a large amount of in-domain data with the current SMT model and keep only those phrase pairs that were frequently used for the final translation, or alternatively considered during the decoding process. They report that they are able to prune about 50% of the phrase-table without any loss in translation quality. The work presented in the current chapter resembles to the work by (Eck et al., 2007) in that the techniques presented here also rely on analysing how likely a certain phrase pair will be used during the translation phase. However, the techniques presented here do not require additional data, but focus on which phrase pairs would be used for generating the current training corpus, in a Viterbi-style training. Furthermore, experimental results show that the techniques presented here are able to yield even larger reductions in phrase-table size.

The work presented in this chapter also relies heavily on the idea of bilingual segmentation. Similarly, (Wuebker et al., 2010) propose the use of a single bilingual segmentation in order to re-estimate translation probabilities by leaving-one-out. As a side effect, the amount of model parameters is also reduced. In the present work, however, the goal of reducing the size of phrase-tables is directly targeted, thus achieving much larger reductions.

2.3 Bilingual segmentation

The problem of segmenting a bilingual sentence pair in such a manner that the resulting segmentation is the one that contains, without overlap, the best phrases that can be extracted from that pair is a difficult problem. First, because of the huge number of possible segmentations that are to be considered. Second, because a measure of optimality must be established. Consider the example:

Source: *La casa verde .*
Target: *The green house .*

When considering this example, one would probably state that a good segmentation for this bilingual pair is *{{La, The}, {casa verde , green house}, {. , .}}*. However, why is such a segmentation better than *{{La , The},{casa verde . , green house .}}*? As humans, we could argue with more or less convincing linguistic terms in favour of the first option, but that does not necessarily mean that such a segmentation is the most appropriate one for SMT. Furthermore, one could possibly think of several linguistically motivated segmentations for this small example.

As described in Chapter 1, a variety of algorithms to extract phrase pairs for SMT have been proposed (Marcu and Wong, 2002; Och and Ney, 2003; Tomas and Casacuberta, 2001; Vogel, 2005). Typically, the bilingual phrases that compose phrase-tables are extracted by using a heuristic algorithm (Zens et al., 2002). Such heuristic algorithm is driven by the following constraint: bilingual phrases must be *consistent* with their corresponding word alignment matrix. However, this process generates huge phrase-tables with highly redundant phrase pairs, since a large number of possible overlapping segmentations are extracted, with the purpose of extracting that segmentation that is useful for the SMT engine. Obviously, such an aggressive approach is bound to be computationally costly, and decoding time greatly suffers because of this issue.

For this reason, the main purpose of this chapter is to reduce the extremely high redundancy in the amount of phrase-pairs that current state-of-the-art SMT systems contain, with the purpose of reducing the time that a human user would be waiting actively for the output to be produced. For doing this, we first examine two different methods to obtain one single segmentation per sentence pair. These two methods rely on the concept of bilingual segmentation. Of course, extracting several overlapping segmentations from a single sentence pair may be beneficial, provided that such segmentations are correct. However, obtaining only the single-best segmentation proves to provide good results, as will be shown in Section 2.7. Nevertheless, obtaining several possible segmentations is also dealt with implicitly in this chapter, in Section 2.6, where the possibility of obtaining n-best segmentations is studied.

In SMT, the concept of phrase-based segmentation entails both the fact of dividing both source and target sentences into phrases, as well as establishing a phrase-based alignment between the phrases obtained. Moreover, such segmentation entails the use of a certain set of bilingual phrases $\boldsymbol{\kappa}$, which are the ones that the decoding algorithm would use to translate a certain input sentence $\boldsymbol{x}$ so as to produce a certain output sentence $\boldsymbol{y}$. We will denote the (ordered) set of phrases used for translating $\boldsymbol{x}$ into $\boldsymbol{y}$ by $\kappa(\boldsymbol{x}, \boldsymbol{y})$, where $\kappa(\boldsymbol{x}, \boldsymbol{y}) \subset \mathcal{B}(\boldsymbol{x}) \times \mathcal{B}(\boldsymbol{y})$, with $\mathcal{B}(\boldsymbol{x})$ and $\mathcal{B}(\boldsymbol{y})$ being the sets of all possible sequences of consecutive words (see Section 1.2.1), of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. In addition, the ordered pairs contained in $\kappa(\boldsymbol{x}, \boldsymbol{y})$ have to include all the words of both the source and target sentences, without overlap. Then, the problem of finding the best segmentation $\hat{\kappa}(\boldsymbol{x}, \boldsymbol{y})$ (or Viterbi segmentation) between $\boldsymbol{x}$ and $\boldsymbol{y}$ can be stated formally as

$$\hat{\kappa}(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\, p(\boldsymbol{\kappa} \mid \boldsymbol{x}, \boldsymbol{y}) \tag{2.1}$$

Operating with this last equation, one easily reach the following, equivalent formulation:

$$\begin{aligned}\hat{\kappa}(\boldsymbol{x},\boldsymbol{y}) &= \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\, p(\boldsymbol{\kappa} \mid \boldsymbol{x},\boldsymbol{y}) \\ &= \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\, \frac{p(\boldsymbol{\kappa},\boldsymbol{y} \mid \boldsymbol{x})}{p(\boldsymbol{y} \mid \boldsymbol{x})} \\ &= \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\, p(\boldsymbol{\kappa},\boldsymbol{y} \mid \boldsymbol{x}) \qquad (2.2)\end{aligned}$$

In addition, it is also possible to reach the last equation by starting from Equation 1.3, describing the typical search process in SMT, which would yield the segmentation $\hat{\kappa}(\boldsymbol{x},\boldsymbol{y})$ as a by-product:

$$\begin{aligned}\hat{\boldsymbol{y}} &= \underset{\boldsymbol{y}}{\operatorname{argmax}}\, p(\boldsymbol{y} \mid \boldsymbol{x}) \\ &= \underset{\boldsymbol{y}}{\operatorname{argmax}} \sum_{\boldsymbol{\kappa}} p(\boldsymbol{\kappa},\boldsymbol{y} \mid \boldsymbol{x}) \qquad (2.3) \\ &\approx \underset{\boldsymbol{y}}{\operatorname{argmax}} \max_{\boldsymbol{\kappa}} p(\boldsymbol{\kappa},\boldsymbol{y} \mid \boldsymbol{x}) \qquad (2.4)\end{aligned}$$

Then, considering the output sentence fixed leads to the same formula as the one presented in Equation 2.2. At this point, three different options for the output sentence $\boldsymbol{y}$ could be considered: the first one, the most obvious one, would be to consider the reference present in the training data, $\boldsymbol{y}^{\tau}$, leading to $\hat{\kappa}(\boldsymbol{x},\boldsymbol{y}^{\tau})$. Alternatively, one could also consider $\hat{\boldsymbol{y}}$, either the one that would be obtained from Equation 2.3 or the one that would be obtained from Equation 2.4, if both do not match, leading to $\hat{\kappa}(\boldsymbol{x},\hat{\boldsymbol{y}})$. Hence, one would suggest that we can perform a search process using a regular SMT system which filters its phrase-table to obtain those translations of $\boldsymbol{x}$ that are compatible with $\boldsymbol{y}^{\tau}$ or $\hat{\boldsymbol{y}}$. Unfortunately, such problem cannot be easily solved, since standard estimation tools such as Thot (Ortiz-Martínez et al., 2005) and Moses (Koehn et al., 2007) do not guarantee complete coverage of sentence pairs seen in training due to the large number of heuristic decisions involved in the estimation process, as described in Section 1.2.1. This means that it is often the case that the SMT system is not able to produce the correct output sentence $\boldsymbol{y}^{\tau}$. In this chapter, two different solutions to this problem are proposed. The first one pursues the goal of obtaining a *true* phrase-based segmentation between $\boldsymbol{x}$ and $\boldsymbol{y}^{\tau}$, whereas the second one focuses on the primary goal of this work, i.e. reducing the amount of bilingual phrases derived from each sentence pair, leading to a *source-driven* bilingual segmentation between $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$.

2.4 True bilingual segmentation

As described in the previous section, coverage problems inherent to state-of-the-art SMT systems imply that it is often impossible to obtain the Viterbi segmentation of a given sentence pair. For this reason, a possible way of overcoming such coverage problems is proposed in (Ortiz-Martínez et al., 2008). In their work, the main idea is to consider every source phrase of $\tilde{x}$ as a possible translation of every target phrase of $\tilde{y}$. For this purpose, two main things are needed: first, a general mechanism to assign probabilities to phrase pairs is needed,

regardless if they are contained in the phrase-table or not, and second, a search algorithm that enables efficient exploration of the set of possible phrase segmentations for a sentence pair.

Such mechanism can be implemented by means of the application of smoothing techniques over the phrase-table. As shown in (Foster et al., 2006), well-known language model smoothing techniques can be imported into the PB translation framework, and these can also be applied to obtain a phrase-level segmentation. According to (Ortiz-Martínez et al., 2008), the best smoothing techniques combine a maximum likelihood phrase-based model statistical estimator with a lexical distribution by means of linear interpolation or backing-off. The lexical distribution uses an IBM 1 alignment model (Brown et al., 1993) that allows to decompose phrase-to-phrase translation probabilities into word-to-word translation probabilities. In the experiments presented here, a phrase-based statistical estimator has been combined with a lexical distribution by means of linear interpolation. In addition, (Ortiz-Martínez et al., 2008) also proposes the use of a log-linear model to control different aspects of the segmentation, such as the number of phrases in which the sentences are divided, the length of the source and the target phrases, the re-orderings and so on. This strategy has also been adopted in the present work. Hence, Equation 2.1 can be rewritten as:

$$\hat{\kappa}(\boldsymbol{x}, \boldsymbol{y}^{\tau}) = \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\, p(\boldsymbol{\kappa}, \boldsymbol{y}^{\tau} \mid \boldsymbol{x}) \tag{2.5}$$

where $p(\boldsymbol{y}^{\tau}, \boldsymbol{\kappa} \mid \boldsymbol{x})$ is given, in this case, by smoothed phrase-based model described above.

Although it might seem that Equation 2.5 matches exactly the decoding problem in SMT, this is not so, since the maximisation takes place only over the segmentation, and is subject to the constraint that $\boldsymbol{y}$ is the actual reference sentence given, $\boldsymbol{y}^{\tau}$. Hence, the typical PB SMT model needs to be smoothed, and the search space is altered.

Once the scoring function for phrase pairs has been defined, a search algorithm to find the bilingual segmentations is required. For this purpose, a search strategy based on the well-known *stack-decoding* algorithm (Jelinek, 1969) can be used. The stack-decoding algorithm for SMT attempts to iteratively *expand* partial solutions, called hypotheses, until a complete translation is found. The expanded hypotheses are stored into a stack data structure which allows the efficient exploration of the search space. Since the number of possible alignments for a given sentence pair may become huge, it is necessary to apply heuristic prunings in order to reduce the search space. The stack-decoding algorithm for SMT cannot be directly applied to bilingual segmentation without certain modifications. Specifically, the stack-decoding algorithm for bilingual segmentation executes a modified expansion algorithm that guarantees the efficient exploration of the set of possible bilingual segmentations for a sentence pair. Such heuristic prunings include the limitation of the maximum number of hypotheses that can be stored in the stack and also the maximum length of the target phrases that can be linked to an unaligned source phrase when expanding a partial hypothesis (Ortiz-Martínez et al., 2008).

The bilingual segmentation procedure that has been described above allows us to compute one true segmentation for each sentence pair. Once the segmentations for every sentence pair have been computed, it is possible to build a phrase-table by only taking into account those segments that are contained in the set of true segmentations.

2.5 Source-driven bilingual segmentation

As it has been explained in Section 2.3, computing $\hat{\kappa}(\boldsymbol{x}, \boldsymbol{y}^{\tau})$ according to a given phrase-table is not an easy task. Specifically, a specific source-target segmentation is often impossible to generate due to coverage problems of the phrase-based model. In the previous section it has been shown how to compute a true phrase segmentation between two given sentences. However, such method must bear with the constraint of having the output sentence fixed. Although such restriction seems logical at training time, it should not be underestimated that this will not be the case in translation time, and such restriction may introduce a non-intended bias. The bilingual segmentation technique described in Section 2.4 allows to overcome coverage problems by combining smoothing techniques with an appropriate search algorithm. This is done at the cost of modifying the scoring function used during the search process due to the application of smoothing techniques, and also by introducing new segment pairs. As said in Section 1, phrase-extraction is typically done by a heuristic algorithm, which has proved to provide appropriate bilingual segments, and altering such segments may not be a good idea.

Since the goal is to discard unnecessary segment pairs contained in the phrase-table, an alternative bilingual segmentation technique that obtains *source-driven* bilingual segmentations is proposed, by relaxing the restriction considered in Equation 2.5, leading to

$$\hat{\kappa}(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \underset{\boldsymbol{\kappa}}{\operatorname{argmax}}\, p(\boldsymbol{\kappa}, \hat{\boldsymbol{y}} \mid \boldsymbol{x}) \tag{2.6}$$

with $\hat{\boldsymbol{y}}$ being the output sentence provided by the search algorithm according to the standard search problem in SMT:

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, p(\boldsymbol{y} \mid \boldsymbol{x}) \tag{2.7}$$

Note that, in this case, $\hat{\boldsymbol{y}}$ may not be the true optimal output sentence according to the translation model, but only the best sentence found by the decoder, which may not match with the true optimal output sentence due to heuristic decisions, approximations and pruning steps performed within the decoder.

Hence, the output sentence $\boldsymbol{y}$ is allowed to be different from the true reference, and the segmentation has been induced by taking into account only the input sentence. By using $\hat{\kappa}(\boldsymbol{x}, \hat{\boldsymbol{y}})$ instead of $\hat{\kappa}(\boldsymbol{x}, \boldsymbol{y}^{\tau})$, we ensure that only segments present in the current phrase-table are used, and no new segments are introduced.

The maximisation described in Equation 2.6 is exactly the same problem as the one of finding the best translation of a source sentence within a phrase-based system, where the segmentation is obtained as a by-product. Hence, for computing $\hat{\boldsymbol{\kappa}}$ it is only necessary to translate each source training sentence and include into the phrase-table those phrase pairs that compose the output hypothesis. Certainly, translating the source sentence does not necessarily produce the target sentence in the training pair, but on the other hand no artificial bilingual segments will be introduced into the phrase-table. In addition, as shown in Section 2.7, experiments show that this approach might be good enough to prune the phrase-table without a significant loss in translation quality.

2.6 Phrase-table pruning and parameter re-estimation

In this section, the source-driven segmentation is generalised. However, to understand the idea behind this generalisation it is key to forget about the bilingual segmentation concept, and consider the source-driven segmentation technique as a full parameter re-estimation method, in which the probability of the phrase pairs is re-computed as the expected number of times that such phrase would be used in translation time. In addition, such probability may also be re-estimated according to the expected quality of the translation generated using that specific phrase pair. To this end, it must be noted that, on the one hand, only parameters (i.e. phrase pairs) that previously had a score greater than zero may yield a score greater than zero after the re-estimation (i.e. no new phrase pairs may appear during the source-driven re-estimation process). On the other hand, phrase pairs which had a score greater than zero may now yield zero score if such phrase pair is never used during the source-driven segmentation, which is the key towards phrase-table pruning.

To state the problem more formally, let $\mathcal{T}$ be a set of training data and $\mathcal{M}$ a SMT model estimated on $\mathcal{T}$. Then the re-estimation technique works as follows:

1. Obtain a set of good translations $G(\boldsymbol{x})$ for each source sentence $\boldsymbol{x} \in \mathcal{T}$ using SMT model $\mathcal{M}$.

2. Extract the set of phrase pairs used to generate all translations $\boldsymbol{y} \in G(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{T}$, using the phrase alignments provided by $\mathcal{M}$.

3. Score each phrase pair according to the number of times such phrase pair has been used.

Here, $G(\boldsymbol{x})$ is defined following two criteria: on the one hand, translations in $G(\boldsymbol{x})$ are selected according to the score assigned by SMT model $\mathcal{M}$; on the other hand, in training time we do have the reference translation $\boldsymbol{y}^{\tau}$, and hence set $G(\boldsymbol{x})$ can be chosen according to a translation quality metric $\mu(\boldsymbol{y}^{\tau}, \boldsymbol{y})$.

Having defined $G(\boldsymbol{x})$ and after obtaining the set of phrases used when generating $G(\boldsymbol{x})$, the probability of each phrase pair $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{x}})$ is re-estimated according to the number of times it was used, weighted by the quality of the hypothesis it appeared in. Hence, segments likely to be used often and within good quality translations obtain higher probability. Formally:

$$p(\tilde{y}|\tilde{x}) \approx \frac{\sum_{\boldsymbol{x} \in T} \sum_{\boldsymbol{y} \in G(\boldsymbol{x})} c(\tilde{x}, \tilde{y}|\boldsymbol{x}, \boldsymbol{y}) \cdot q(\boldsymbol{y})}{\sum_{\tilde{y}'} \sum_{\boldsymbol{x} \in T} \sum_{\boldsymbol{y} \in G(\boldsymbol{x})} c(\tilde{x}, \tilde{y}'|\boldsymbol{x}, \boldsymbol{y}) \cdot q(\boldsymbol{y})} \tag{2.8}$$

where $c(\tilde{x}, \tilde{y}|\boldsymbol{x}, \boldsymbol{y})$ is the total number of times that phrase pair $(\tilde{x}, \tilde{y})$ is used when translating source sentence $\boldsymbol{x}$ into hypothesis $\boldsymbol{y}$, and $q(\boldsymbol{y})$ is a weighting factor which accounts for how good does $\boldsymbol{x}$ translate into $\boldsymbol{y}$. Three different approaches are analysed:

1. $q(\boldsymbol{y}) = 1$: assume that the probability of a phrase pair does not depend on the quality of the hypothesis it has appeared in. This is the standard approach to score segments in state-of-the-art SMT systems (Zens et al., 2002).

2. $q(\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})$: assign each phrase pair a weight given by the likelihood of translating $\boldsymbol{x}$ into $\boldsymbol{y}$ according to SMT model M.

3. $q(\boldsymbol{y}) = \mu(\boldsymbol{y}, \boldsymbol{y}^{\tau})$: assign each phrase pair a weight given by quality metric $\mu(\cdot,\cdot)$. Since we are generating $G(\boldsymbol{y})$ with M, the translations provided may differ from the reference translation $\boldsymbol{y}$, which is given in training time. Hence, we can assess the real quality of hypothesis $\boldsymbol{y}$. This implies that a given phrase pair will be weighted according to the expected quality of the sentence it appears in.

Although Equation 2.8 implies a re-estimation of the translation parameters, it must be noted that it also implies an aggressive pruning in the amount of parameters present in the translation model, i.e. the number of phrases in the phrase-table: since the estimation of $p(\tilde{y}|\tilde{x})$ is based on a set $G(\boldsymbol{x})$ of good translations of $\boldsymbol{x}$, only those phrases present in such translations will be assigned a probability greater than zero, and the rest will be pruned out. Although it might seem that a smoothing step is needed, the goal is actually to prune those phrase pairs that do not seem to be useful, and experimental results show that, in fact, such smoothing is not necessary. In this way, a phrase-table containing only those segments likely to be used in translation time is obtained. Note, however, that the smoothing mentioned here implies smoothing the probabilities of existing phrase pairs, and its effects are completely different from those introduced by smoothing in the case of the true segmentation strategy.

2.7 Experimental results

Experiments on this subject will be conducted by means of the Thot and Moses toolkits. On the one hand, the experiments concerning both bilingual segmentation techniques will be conducted by means of Thot, since this toolkit includes a tool for segmenting both input and output sentences following the true segmentation strategy. Hence, for comparison reasons, the source-driven technique will also be performed by means of the Thot toolkit. On the other hand, once these experiments were performed and the potential of both techniques was established, the generalisation of the source-driven technique, described in Section 2.3, is analysed by using the more state-of-the-art toolkit Moses, with the purpose of providing results which could be comparable with those provided in recent SMT evaluation campaigns. The corpora used will be the Europarl corpus (see Section 1.4).

2.7.1 Bilingual segmentation experiments

Experiments for assessing the effectiveness of the source-driven and true bilingual segmentation techniques were performed by means of the Thot toolkit (see Section 1.5), in its default monotonic setup. Results for the source-driven segmentation strategy are shown in Table 2.1. In addition to the typical BLEU and TER scores, and since the main purpose is to measure computational efficiency, *speedup* (S_p) and phrase-table size are also provided. On the one hand, speedup is defined as

$$S_p = T_b/T_r \tag{2.9}$$

where T_b is the time taken by the baseline system and T_r is the time taken by the system with the reduced phrase-table. On the other hand, phrase-table size is presented in millions

	Baseline				Source-driven					
Pair	BLEU	TER	size	w/s	BLEU	TER	size	w/s	size red.	S_p
Es–En	28.2	56.0	5.0	93	27.5	56.2	0.05	1500	99.0%	16
En–Es	27.6	56.6	5.1	76	27.2	56.6	0.12	700	97.6%	9
De–En	21.6	64.8	4.2	100	21.1	64.8	0.06	1500	98.6%	15
En–De	15.2	70.9	5.5	46	15.1	70.2	0.14	400	97.5%	9

Table 2.1: Translation quality, number of model parameters measured in terms of millions of phrase pairs, number of translated words per second and speedup (S_p) obtained when using a PB translation system for the source-driven segmentation technique. Monotonic search was considered. PB model size is given in millions of phrase-pairs.

of phrase-pairs, measured after filtering the phrase-table according to the current test set, as is typically done when the test set is available beforehand because loading the complete phrase-table without any kind of filtering is usually unfeasible even with modern machines.

Since these experiments are somewhat older, they were conducted on the Europarl corpus, in the partition established for the WMT07 Workshop (see Section 1.4). The development set was used for estimating the weights $\boldsymbol{\lambda}$ of the log-linear combination and the test set was used for evaluation purposes. Note that, since the *Test* set was used, and not the *Devtest* set, the evaluation data contains an out-of-domain subset, which implies that the problem of reducing the phrase-table is even more challenging because the proposed techniques need to avoid the possible over-fitting that such reduction could entail.

Results for the source-driven segmentation technique can be seen in Table 2.2. As shown, translation quality is not significantly affected by the reduction of the size of the phrase-table proposed. On the one hand BLEU scores are slightly lower than those of the baseline system, although confidence tests conducted by means of *Test-specific* bootstrap re-sampling (see Section 1.2.2) showed that these differences are not statistically significant. On the other hand, TER scores seem to remain completely unaltered, even though a very slight variation can be observed (0.2 worse for Es–En, 0.7 better for En–De).

As for the number of parameters of the models used, it can be seen that such number is reduced in two orders of magnitude, i.e. the number of parameters remaining in the phrase-table after applying the source-driven technique is only around 2% the original number of parameters. Moreover, translation speed is increased by a factor between 9 and 16, all this without a significant loss in translation quality. In addition, given that the resulting phrase-table is much smaller, it would be possible to fit the complete phrase-table (i.e. without test-specific filtering) into memory, which implies that the SMT system could be set online for translation without the need of knowing the test set in advance or using phrase-table binarisation techniques.

Results for the true segmentation strategy are shown in Table 2.2. As opposed to source-driven segmentation, translation quality does drop significantly (although not consistently) with respect to the baseline, ranging from 0.5 to 4.4 BLEU points and from 0.2 to 5.1 TER points. In addition, the reduction in size is slightly smaller than in the case of the source-driven segmentation, and it also seems that the segments kept introduce quite some ambiguity,

since speedup is significantly lower than in the former case.

One key difference between the two proposed techniques consists in the degree of similarity of the pruned phrase-tables obtained by the techniques with respect to the original phrase-table. Although the true bilingual segmentation allows to obtain a complete segmentation of the source and target sentences, this comes at the cost of introducing smoothing techniques. Hence, the resulting segmentations contain phrase pairs that are not present in the original phrase-table. In the experiments carried out, the pruned phrase-tables generated by the true bilingual segmentation contained a relatively high number of phrase pairs that were not present in the original phrase-tables, ranging from 10% to 50% depending on the language pair. In contrast, the source-driven bilingual segmentation, since it merely consists in translating the source sentence, always generates a pruned phrase-table that is a true subset of the original phrase-table. This suggests that the true segmentation technique not only prunes the original phrase-table, but also has an important role in the estimation of new model parameters, which could be the reason for the degradation of the translation quality.

	Baseline				True					
Pair	BLEU	TER	size	w/s	BLEU	TER	size	w/s	size red.	S_p
Es–En	28.2	56.0	5.0	93	23.8	60.8	0.07	380	98.6%	4
En–Es	27.6	56.6	5.1	76	24.7	60.1	0.16	250	96.9%	3
De–En	21.6	64.8	4.2	100	17.5	69.9	0.22	280	94.8%	3
En–De	15.2	70.9	5.5	46	14.7	71.1	0.31	170	94.4%	4

Table 2.2: Translation quality, number of model parameters, number of translated words per second and speedup (S_p) obtained when using a PB translation system for the true segmentation technique. Monotonic search was considered. PB model size is given in millions of phrase-pairs.

2.7.2 Parameter re-estimation experiments

Once it was established that the source-driven segmentation technique works properly for pruning the phrase-table, such technique was considered as a pure parameter re-estimation method, as described in Section 2.6. In this case, the Moses toolkit was used for the experiments, with the purpose of providing state-of-the-art results that could be compared with those presented in recent SMT evaluation campaigns and since the comparison between the source-driven and true segmentation techniques has already been established. In addition, the more recent version of the Europarl corpus was used, i.e. the partition of the corpus established in the WMT10 Workshop (Section 1.4). The test set used for evaluation purposes was, as in the previous section, the *Test* subset (see Section 1.4).

As for the $G(\cdot)$ and $q(\cdot)$ functions described, three settings are analysed:

1. $q(\boldsymbol{y}) = 1$ and $G(\boldsymbol{x})$ chosen according to the order in the n-best list provided by the SMT model. This approach is equivalent to the original source-driven segmentation strategy, when using only the first-best hypothesis. This setting will be referred to by `flat`.

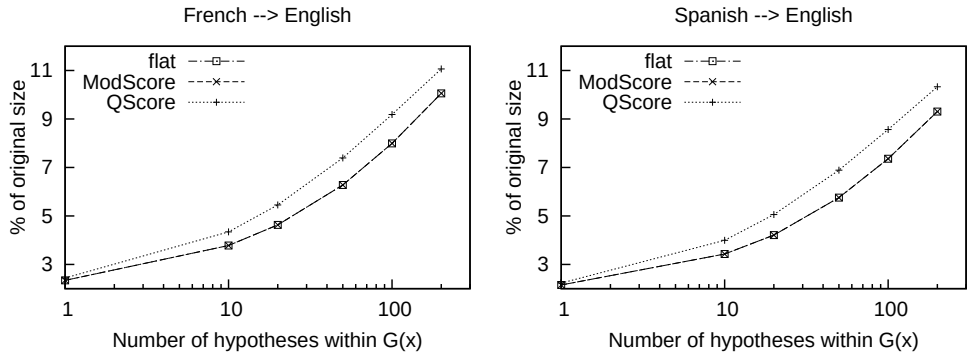


Figure 2.1: Amount of phrases present in the reduced system, given as % with respect to the original system.

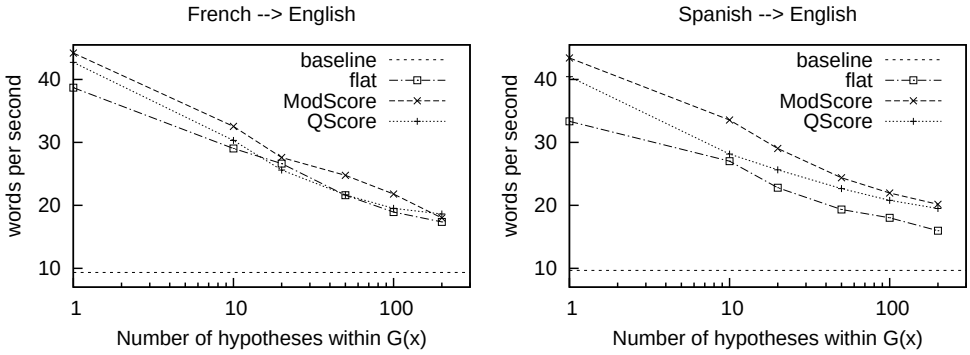


Figure 2.2: Decoder speed for original and filtered systems.

2. $q(\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})$ and $G(\boldsymbol{x})$ chosen as above. This setting will be referred to by `ModScore`.

3. $q(\boldsymbol{y}) = \mu(\boldsymbol{y}, \boldsymbol{y}^{\tau})$ and $G(\boldsymbol{x})$ chosen according to the order defined by quality metric $\mu(\cdot,\cdot)$. This setting will be referred to by `QScore`.

The effect of applying the different settings described above was studied. The amount of phrases present in the phrase-table for each of the settings described is shown in Figure 2.1. The method implemented achieves a reduction of about 97% in the amount of phrases present in the phrase-table, without a significant loss in translation quality, yielding a SMT system that is able to fit into portable devices: when considering only the first-best hypothesis, the size of the phrase-table that the decoder had to load into memory was only about 14MB, and about 35MB when including 50 hypotheses into $G(\boldsymbol{x})$, versus 450MB for the original system. Although these sizes were measured after filtering the phrase-table according to the test set, as is usually done in SMT, similar conclusions can be obtained when analysing the complete

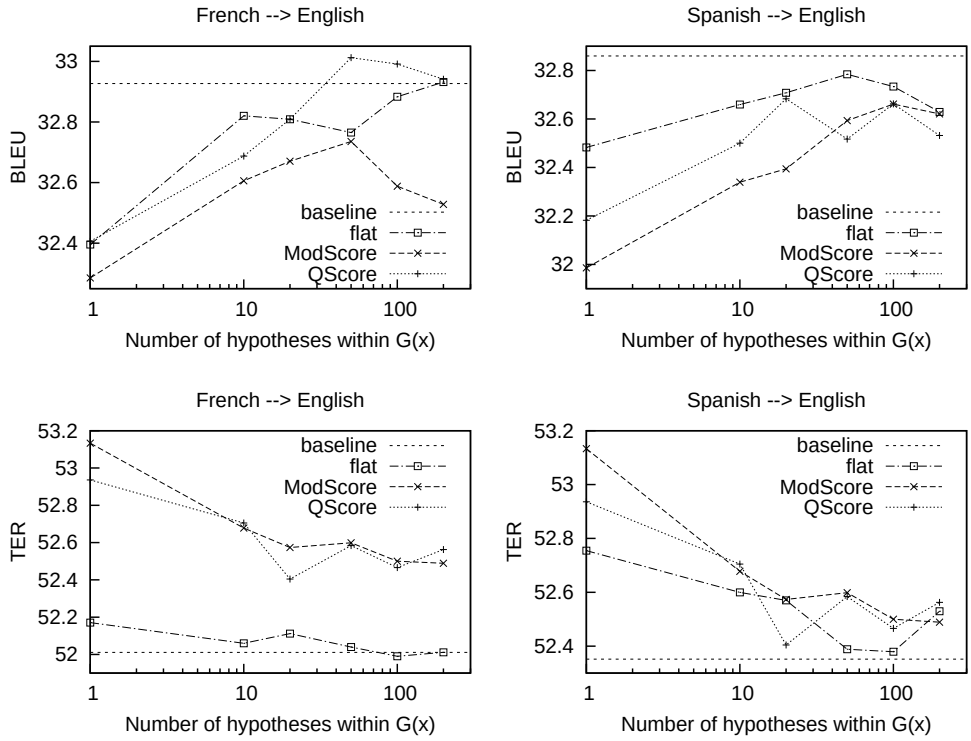


Figure 2.3: Translation quality, as measured by BLEU and TER, for the baseline system and the pruned systems.

phrase-tables. As expected, the phrase-table size increments when considering an increasing size of $G(\boldsymbol{x})$. Settings `flat` and `ModScore` present the same amount of phrases, since both use the same $G(\boldsymbol{x})$. In order to study the impact of phrase-table size reduction, translation speed was measured. As shown in Figure 2.2, the speed that the pruned system is able to deliver increases in a very significant manner, achieving more than three times the original speed. In this plot, it can be seen that the speed of the baseline system is much slower than in the case of the previous subsection, in the experiments regarding the comparison between the source-driven and true segmentation techniques. This is not because Thot is much faster than Moses, but because in the present case lexicalised re-ordering is considered, whereas in the previous case only monotonic decoding was taken into account. In Figure 2.1 it can also be seen that an average sentence of 30 words, as is the case in most of the corpora considered in the present thesis, will take less than one second to translate, even when considering re-ordering, which is perfectly tolerable even with a human translator waiting actively for the translation.

The effect on translation quality of the re-estimation techniques described was also studied, and translation quality results are shown in Figure 2.3. As shown, using only the first-best

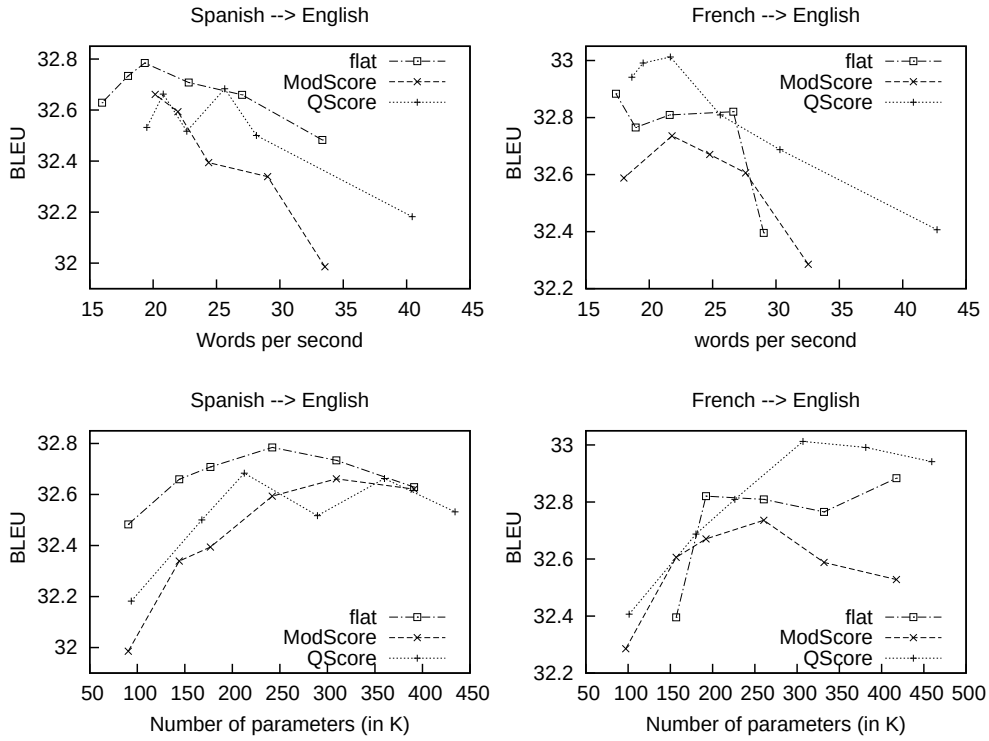


Figure 2.4: Relationship between speed/phrase-table size and the translation quality achieved for the three different strategies analysed.

hypothesis for $G(\boldsymbol{x})$ leads to a slight degradation in translation quality, as was also the case in the experiments with the segmentation techniques. However, when including 10 hypotheses into $G(\boldsymbol{x})$, this difference is already very scarce, and increasing the size of $G(\boldsymbol{x})$ yields SMT systems that are able to deliver the same translation quality as the original system, for all the settings analysed. Another thing that can be noted is that setting `QScore` appears to perform better than the other settings considered, which seems reasonable since $G(\boldsymbol{x})$ takes into account the translation quality of a given sentence before including the phrases it is built of. Although it might seem that it is able to improve the baseline in terms of translation quality for French→English, this is not statistically significant, and such finding was not coherent in other language pairs.

So as to illustrate the translation quality that would be expected when required a certain speed, or when having certain memory restrictions, speed and phrase-table size are plotted against BLEU in Figure 2.4. Although, as in the plots, there is no method that clearly performs better (or worse) than the others, it does appear that the `ModScore` setting is the one that performs worse in terms of requirements/translation quality ratio.

However, there appears to be no significant difference between the three settings anal-

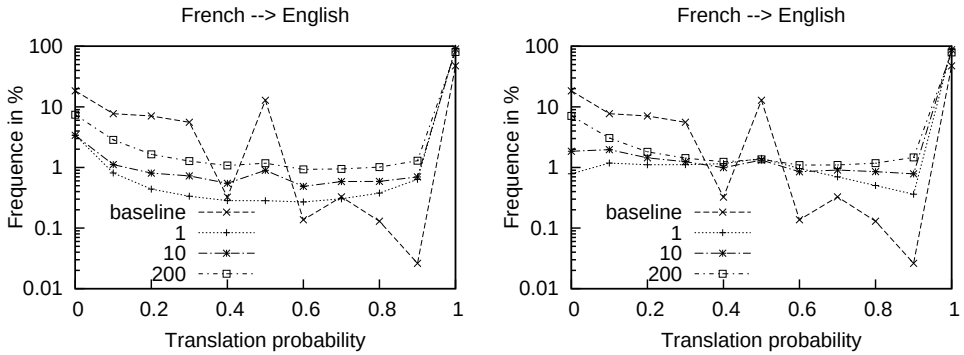


Figure 2.5: Relative frequency for each discretized value of $p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}})$, considering baseline system and different sizes of $G(\boldsymbol{x})$ for the `ModScore` setting (left) and the `QScore` setting (right).

ysed. This can be explained by considering Figure 2.5. For plotting this figure, the direct translation probability $p(\boldsymbol{y}|\boldsymbol{x})$ was rounded to have only one decimal number, and then the relative frequency of each value was plotted. The plots corresponding to the other language pairs studied were almost undistinguishable from the present ones (even concerning the shape of the baseline system). Note that the y-axis is in logarithmic scale for visibility purposes. As shown, the original system presents a relatively large number of phrase pairs with low probabilities: for about 35% of the phrase pairs, $p(\boldsymbol{y}|\boldsymbol{x}) < 0.4$. However, in the reduced system, less than 10% of the phrase pairs have a probability lower than 0.95. In fact, in the case of considering only the first-best hypothesis, $p(\boldsymbol{y}|\boldsymbol{x}) = 1$ for about 84% of the phrase pairs, versus 47% for the original system. This means two things: on the one hand, that the actual choice for $q(\cdot)$ will have a limited effect, since it will only affect 16% of the phrase pairs, although such statement does not necessarily need to hold for the selection function $G(\cdot)$. On the other hand, that in most cases a certain source phrase will be associated with a single target phrase, and the only decisions that the decoder will need to take regard how to segment the source sentence and then re-order target phrases. Observing Figure 2.5, it could be argued that a faster technique for phrase-table pruning could be to keep only those phrases that have $p(\boldsymbol{y}|\boldsymbol{x}) = 1$. However, such strategy leads to a phrase-table of 17% the original size, and a BLEU score of 12, i.e. larger phrase-tables and much worse translation quality.

One last note regards average phrase length. Although it is reasonable to think that the reduced systems will tend to keep longer phrases, this issue is mitigated by the fact that phrase length is also a feature considered within state-of-the-art SMT systems, and its weight is adjusted by the MERT procedure according to a given development set. In this sense, it was observed that the pruned phrase-tables presented slightly longer phrases, although the difference was never above 14% in the experiments detailed in this section. The difference in average phrase length seemed to depend more on the size of $G(\cdot)$ than on $q(\cdot)$, and including more segmentations per sentence tended to yield shorter average phrase lengths.

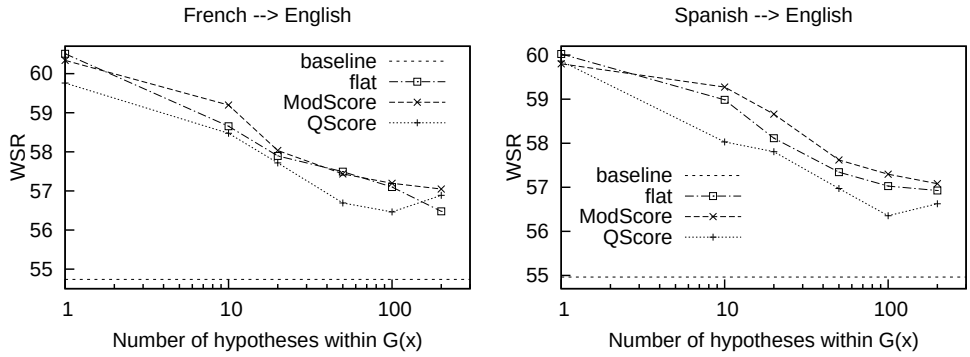


Figure 2.6: WSR achieved when applying the parameter re-estimation techniques detailed above, for French→English and Spanish→English translation. `flat`, `ModScore` and `Qscore` are the settings defined in the previous sub-section. Confidence intervals are not shown for clarity reasons, but their size was always between 1.64 and 1.90.

2.7.3 Interactive machine translation results

In addition to the experiments conducted in the SMT framework, additional experiments were conducted with the purpose of assessing whether the parameter re-estimation technique presented here provides equivalent results in IMT. For doing this, the PB SMT systems developed in the previous subsection were employed for producing word-graphs, and these were then used as back-end for the IMT system. The results in terms of WSR for this experimentation can be seen in Figure 2.6. As shown, when applying the parameter re-estimation technique described in Section 2.6, the reduced systems present a lower performance than the baseline system, as measured by WSR. However, it should be noted that this difference is only statistically different when the size of $G(\boldsymbol{x})$ is smaller than 20. In addition, it also seems that the `QScore` setting is the one that yields the best performance compared to the other reduced systems. Nevertheless, even though this observation seems to be mostly true in the experiments performed, the differences are not statistically significant.

In terms of the time required by the system to produce its output, Figure 2.7 shows two different comparisons. The upper two plots display the total average time required by the system to produce the final output. As shown in the plots, the baseline system is about three times slower than the reduced systems, when setting the size of $G(\boldsymbol{x})$ to 1, and about 50% slower when the size of $G(\boldsymbol{x})$ is set to 200. At this point, it should be remembered that this time is computed by simulating the user, i.e. by assuming that the user would want to produce exactly the same sentence present in the reference, and also by assuming that interaction of the user takes no time at all. In addition, the total time taken also depends on the number of interactions simulated, i.e., the total number of times that a suffix had to be produced. For this reason, the average time taken by each system to produce a suffix hypothesis was also measured, and these are the results shown in the two bottom plots of Figure 2.7. In this case, the reduced systems perform about three times faster than the baseline system when

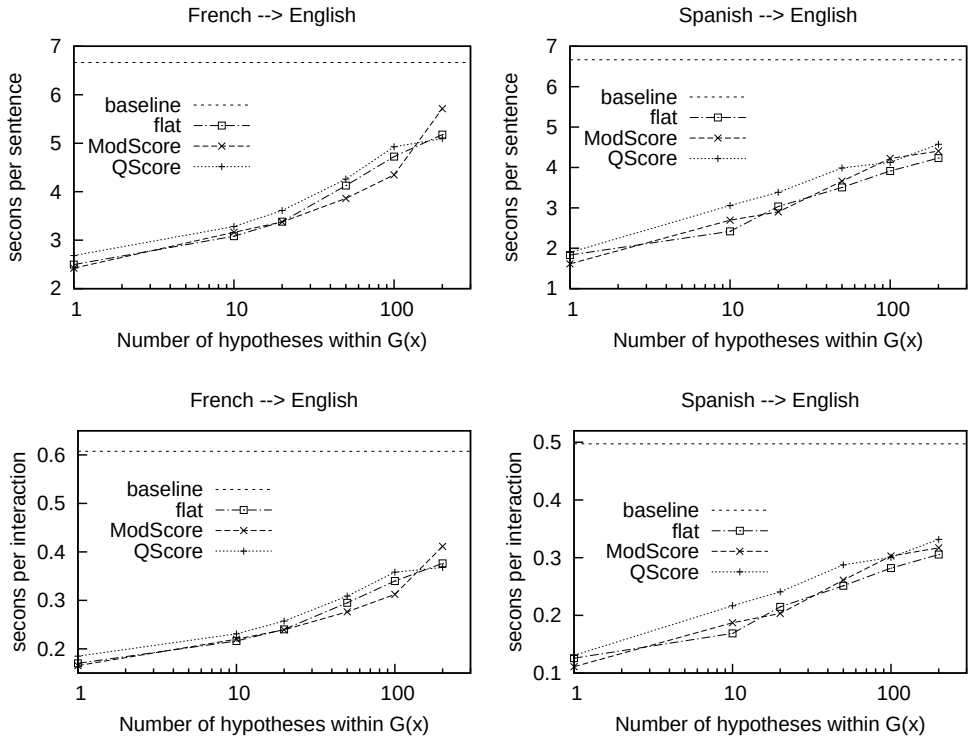


Figure 2.7: Temporal evaluation of the re-estimation techniques detailed above, for French→English and Spanish→English translation. The two plots on the top show how many sentences were produced per second in a user-simulated environment, while the two plots on the bottom show the time consumed in average to produce one single suffix-hypothesis.

$|G(\boldsymbol{x})| = 1$, and about twice as fast as the baseline when $|G(\boldsymbol{x})| = 200$. Although this plot is more meaningful, since it shows the average time taken by the system to respond after a given interaction of the user, there is still one aspect which makes these plots not totally clear: when the suffix hypothesis to be produced is the whole translation, the IMT system takes much more time than when the suffix to be produced is just some words long. However, the speed gains achieved by applying the pruning strategies described in the present chapter do not seem to depend much on the length of the suffix to be produced. In the worst of the cases, the suffix to be produced is the whole sentence, which is the same case as in the SMT experiments. Since the speedup achieved in SMT is coarsely similar to the one achieved in IMT on a per suffix basis, it can be concluded that the computational gain does not depend on the length of the suffix to be produced. Nevertheless, in absolute terms, the system will take much more time to produce longer suffices, and hence the need of pruning techniques will be more evident.

2.8 Conclusions and future work

In the present chapter, a technique for pruning the phrase-table is presented. Such technique relies mainly on the concept of bilingual segmentation, although a generalisation may turn it into a parameter re-estimation technique. The technique presented attempts to assess how likely is it for a given phrase pair to be used in translation time, and discard it whenever it is too unlikely to be used. In an attempt to promote those segments which appear in good quality translations, the resulting phrase pairs may be weighted by the quality of the sentence produced.

Four main conclusions are drawn. First, that it is possible to reduce the phrase-table by 97% without any significant loss in translation quality, yielding a decoding speed of about four times faster the original speed, making it possible to use a PB SMT system in a real-time environment where a human translator is waiting actively. Given that the translation model obtained is much smaller, the presented technique is also adequate for integrating SMT systems into hand-held devices without the need of sacrificing translation quality.

Second, that the true segmentation technique does not seem to be an appropriate phrase-table reduction technique. This is most probably because the smoothing needed to compensate for the coverage problems present in PB SMT systems forces the introduction of too many new phrase pairs, which may not be the most adequate.

Third, that the amount of phrase pairs present in the phrase-table after the source-driven segmentation technique (or the specific approaches derived from its generalisation) has been applied is already very close to the minimum set of phrases that are needed within the phrase-table if no degradation in translation quality is desired. This is evidenced by the fact that an important amount of the resulting phrase pairs are assigned probability 1 in the different SMT models (i.e. feature functions) present. Hence, performing a re-estimation of the parameters may not be able to yield positive results at all, since the resulting phrase-table is already almost deterministic.

Lastly, experimental results concerning IMT show that the word-graphs produced by the pruned systems are not as rich as the ones produced by the baseline system. For this reason, a human translator would need to perform more interactions in order to correct the initial hypothesis in the case of the pruned systems. However, such increase might be welcome whenever the sentences to be translated are sufficiently long, with the purpose of having the system respond in real time. In this sense, one possible extension to the present work would be to use the pruned system only with the purpose of generating the first translation hypothesis, and then use the word-graphs provided by the baseline systems to produce the successive suffices, in the spirit of improving the time taken by the system only in those cases where such time is critical.

An inmediate direction for extending the work presented here is to develop other smoothing strategies for the true segmentation technique. In this sense, it is reasonable to assume that the result achieved by the source-driven approach should constitute a sort of lower bound for the true approach, i.e. the purpose should be to achieve with the true segmentation *at least* the same results obtained with the source-driven segmentation.

Another topic which still deserves a deeper analysis is the definition of $G(\cdot)$. It appears that considering different $q(\cdot)$ functions does not have an important effect on the final translation quality achieved, since the resulting phrase-table has a very low ambiguity. However,

exploring further options for $G(\cdot)$ may still present a promising extension.

The source-driven segmentation strategy presented in this chapter was first published in an international workshop:

- **G. Sanchis-Trilles** and F. Casacuberta. Increasing Translation Speed in Phrase-Based Models via Suboptimal Segmentation. In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems, PRIS 2008*, pages 135–143, INSTICC Press, Barcelona (Spain), June 2008.

The source-driven segmentation strategy also lead to a publication in an international conference, by applying it for building Finite State Transducers:

- J. González, **G. Sanchis-Trilles** and F. Casacuberta. Learning Finite State Transducers Using Bilingual Phrases. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2008*, pages 411–422, Lecture Notes in Computer Science, Haifa (Israel), February 2008.

Finally, the comparison between both source-driven and true strategies was published in an international conference:

- **G. Sanchis-Trilles**, D. Ortiz-Martínez, J. González-Rubio, J. González and F. Casacuberta. Bilingual segmentation for phrasetable pruning in Statistical Machine Translation. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation, EAMT 2011*, pages 257–264, Leuven (Belgium), May 2011.

The experimental results achieved by generalising the source-driven segmentation strategy are not yet published, although an article is being prepared for submission to an international conference.

Bibliography

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of machine translation. *Computational Linguistics*, 19(2): 263–311, 1993.

Matthias Eck, Stephan Vogel, and Alex Waibel. Translation model pruning via usage statistics for statistical machine translation. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 21–24, April 22–27 2007.

George Foster, Roland Kuhn, and Howard Johnson. Phrasetable smoothing for statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 53–61, July 22–23 2006.

Frederick Jelinek. Fast sequential decoding algorithm using a stack. *IBM Journal of Research Development*, 13(6):675–685, 1969.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. Improving translation quality by discarding most of the phrasetable. In *Proceedings the conference on Empirical Methods in Natural Language Processing and Computational Natural Language Processing*, pages 967–975, June 28–30 2007.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics: Demo and Poster Sessions*, pages 177–180, June 23–30 2007.

Daniel Marcu and Daniel Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 133–139, July 6–7 2002.

Franz J. Och and Hermann Ney. Improved statistical alignment models. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 440–447, October 1–8 2000.

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

D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. The scaling problem in the pattern recognition approach to machine translation. *Pattern Recognition Letters*, 29(8):1145–1153, 2008.

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. Thot: a toolkit to train phrase-based statistical translation models. In *Proceedings of the Machine Translation Summit X*, pages 141–148, September 12–16 2005.

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. Phrase-level alignment generation using a smoothed loglinear phrase-based statistical alignment model. In *Proceedings of the conference of the European Association for Machine Translation*, pages 158–167, September 22-23 2008.

Jesús Tomas and Francisco Casacuberta. Monotone statistical translation using word groups. In *Proceedings of the*

Machine Translation Summit X, pages 357–361, September 18–22 2001.

Stephan Vogel. PESA: Phrase Pair Extraction as Sentence Splitting. In *Proceedings of the Machine Translation Summit X*, pages 251–258, September 12–16 2005.

Joern Wuebker, Arne Mauser, and Hermann Ney. Training phrase translation models with leaving-one-out. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 475–484, July 11–16 2010.

Richard Zens, Franz J. Och, and Hermann Ney. Phrase-based statistical machine translation. In *Proceedings of Advances in Artificial Intelligence: the annual German conference on Artificial Intelligence*, pages 18–32, September 16–20 2002.

CHAPTER 3

Language model adaptation for statistical machine translation

Someone will do it. We have to be that one.

Marcello Federico

Contents

3.1 Introduction **55**
3.2 Related work **55**
3.3 General framework for language model adaptation **56**
3.4 Supervised labelled data for language model adaptation **58**
3.5 Unsupervised clustering for language model adaptation **62**
3.6 Weight optimisation strategies **64**
3.7 Experimental results **65**
3.8 Conclusions and future work **78**
Bibliography **80**

He felt faint again now but he held on the great fish all the strain that he could. I moved him, he thought. Maybe this time I can get him pull over. Pull, hands, he thought. Hold up, legs. Last for me, head. Last for me. You never went. This time I'll pull him over.

[...]

"I wish I had a stone for the knife," the old man said after he had checked the lashing on the oar butt. "I should have brought a stone." You should have brought many things, he though. But you did not bring them, old man. Now is no time to think of what you do not have. Think of what you can do with what there is.

The Old Man and the Sea. Ernest Hemingway.

Se sentía débil ahora de nuevo, pero él llevó a cabo en el gran pez toda la tensión que podía. Lo movía, pensó. Quizás esta vez pueda conseguir que se detuviera. Pull, las manos, pensó. Levante las piernas. Última para mí, con la cabeza. Última para mí. Que nunca fue. Esta vez lo voy a tirar encima.

[...]

"Me gustaría tener una piedra por el cuchillo", dijo el anciano después de haber comprobado los azotes en el culo remo. "Tendría que haber traído una piedra." Usted debería haber traído muchas cosas, sin embargo. Sin embargo, no ha traído, viejo. Ahora no es momento de pensar en lo que no tienen. Piense en lo que se puede hacer con lo que hay.

El viejo y el mar. Google Translate.

3.1 Introduction

In this chapter, the problem of language model adaptation as applied to statistical machine translation is examined. In this context, n-gram mixtures of language models are investigated, which are obtained by clustering bilingual training data. Several clustering techniques are analysed, some of them attempting to exploit existing manually-annotated information, others researching different ways of clustering the training data automatically in an unsupervised manner. Then, in translation time, the mixture weights are estimated at several degrees of granularity, ranging from the pure sentence level to weights estimated on the complete test set. Experimental results show that, by training different specific language models weighted according to the actual input instead of using a single target language model, translation quality improvements can be achieved, both in terms of BLEU and in terms of TER.

Hence, the purpose of this chapter is to study different ways of augmenting the LM component of the SMT system by introducing parameters that are adapted dynamically to the input text. With this purpose, the LM is implemented as a mixture of specialised sub-LMs, which are conveniently estimated through some bilingual clustering of the training data and then combined following different weighting schemes.

Most part of the work detailed in this chapter was carried out during a 3-month internship at the *Fondazione Bruno Kessler* in Trento, Italy, in collaboration with Dr. Marcello Federico and Mauro Cettolo. The author of this thesis is very grateful to both of them for granting him such an opportunity.

This chapter is organised as follows. Section 3.2 briefly lists other works dealing with related issues, both regarding LM adaptation in SMT and other related fields, and also regarding the use of mixture models for adaptation in SMT. The general framework for LM adaptation by means of n-gram mixtures researched in this chapter is described in Section 3.3. Section 3.4 describes the different supervised approaches studied when dealing with the clustering problem. Then, in Section 3.5, different unsupervised clustering approaches are analysed for the case where no manually annotated data exists. Different strategies for assigning the n-gram mixture weights are described in Section 3.6. The experimental results obtained by means of these procedures are described in Section 3.7, and the conclusions which can be drawn from the present work are described in Section 3.8. This last section also describes the future work still to be done.

3.2 Related work

One of the first approaches to adaptation in SMT was proposed by (Lagarda and Juan, 2003), in which the translation model is implemented as an unsupervised multinomial mixture of translation models, where each component is supposed to concentrate most of its probability mass on a certain topic. Mixture models for adaptation were also explored in (Civera and Juan, 2007). However, in this case the mixtures were designed for word alignment modelling. With this purpose, the authors proposed to replace the standard word-alignments by mixtures of the HMM alignment model. Since mixture modelling induces soft partitions, topic specific alignments were defined by each mixture component. Although achieving interesting improvements in terms of alignment error rate, improvements in terms of translation quality

were more limited given the large amount of heuristics applied after the word-alignment step in order to extract phrases.

Slightly later, (Nepveu et al., 2004) applied other adaptation techniques to interactive MT, following the ideas in (Kuhn and Mori, 1990) and adding cache language and translation models to their system. Following the same concept present in hardware cache memories, the purpose of TM and LM caches is to track short-term fluctuations in word (or phrase pair) frequency. Then, these caches are combined in a log-linear fashion with the generic LM and TMs. Although the language model caches did produce interesting improvements in terms of translation quality, translation model caches did not seem to provide further improvements.

Other authors followed a different approach when confronting the adaptation problem. For instance, (Koehn and Schroeder, 2007) studied different ways to combine in-domain data with out-of-domain data. Their experiments ranged from the simple concatenation of all data available to more complex combination strategies, such as establishing different translation and language models which were combined in a log-linear fashion. In a conceptually similar work, (Bertoldi and Federico, 2009) also explored different ways to combine in-domain and out-of-domain data, although in this case the data added is only source language data.

Language model adaptation has been deeply explored since at least the mid 90s in the ambit of speech recognition (Bellagarda, 2001; Mori and Federico, 1999). Nowadays, also in the SMT community the interest for LM adaptation is continuously growing. More specifically, there has been a recent effort towards providing the SMT system with a more adaptable LM. For example, (Zhao et al., 2004) propose to build a query from a list of candidate translations for each source sentence. Such query is used to retrieve similar sentences from a very large training corpus, and the sentences retrieved are used to build specific LMs which are then interpolated in translation time with a background LM estimated on all the data available. Finally, the source sentence is re-translated by using the interpolated LM. By doing this, they report that they are able to provide stable, although very limited improvements over the single-LM baseline.

Similarly, (Lü et al., 2007) propose to use *term frequency-inverse document frequency* (TF-IDF) to select similar data within the same training corpus, and then prepare specific LMs and TMs. These specific models are then interpolated in translation time according to different weighting schemes. As in the case of (Zhao et al., 2004), they also report minor but stable improvements in translation quality metrics. In a similar work, (Yamamoto and Sumita, 2007) propose to cluster the bilingual training corpus so as to minimise the entropy of each subset, and then train independent language and translation models from these smaller bilingual corpora, which are in turn recombined in translation time by performing domain prediction. Differently, in the present work the final combination of target LMs is obtained by re-using the weights estimated by maximising the probability of generating the source sentence by means of the linear interpolation of source sub-LMs.

3.3 General framework for language model adaptation

The key idea behind the language model adaptation technique presented in this chapter consists in replacing the language model present in Equation 1.8, which is one of the feature functions $h(\cdot,\cdot)$. Specifically, such feature is typically the language model of the output sen-

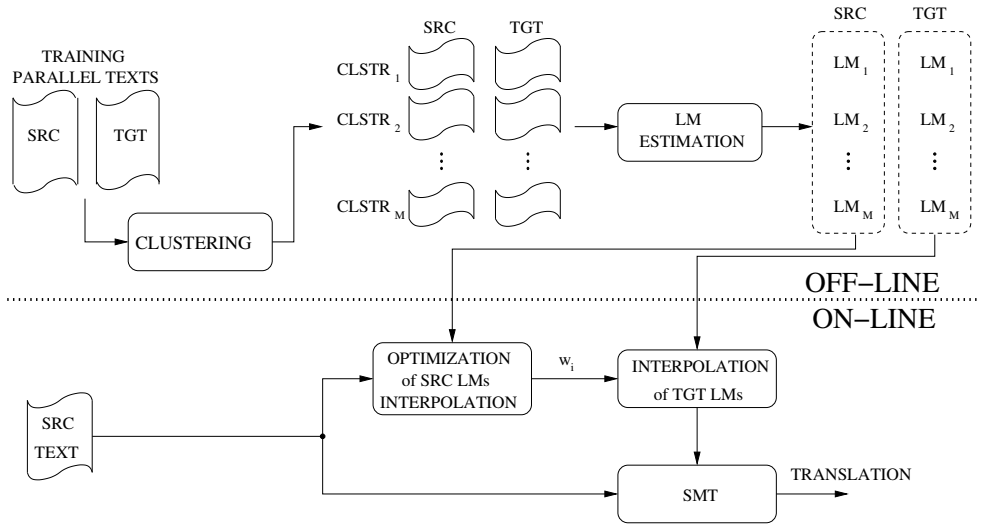


Figure 3.1: Basic procedure for LM adaptation.

tence, i.e.

$$h(\boldsymbol{x}, \boldsymbol{y}) = \log p(\boldsymbol{y}) \tag{3.1}$$

which provides the logarithm of the probability assigned by the target LM to the output sentence $\boldsymbol{y}$. Typically, this probability is most often given by a single word-based LM. In this work, this formula is extended by considering that such probability is given by a linear interpolation (mixture) of word-based language models, i.e.

$$p(\boldsymbol{y}) = \sum_{i=1}^{M} w_i p_i(\boldsymbol{y}) \tag{3.2}$$

where each $p_i(\boldsymbol{y})$ is a LM trained on sentences of the target language. However, considering the final probability $p(\boldsymbol{y})$ as a linear interpolation allows the introduction of several different language models, which may be estimated from different subdivisions of the training data available. With the help of Figure 3.1, the basic procedure for LM adaptation is described in the following. Note that this procedure is thought for adapting a LMs trained on the target side of the parallel corpus in consideration, i.e., LMs trained on other (monolingual) corpora cannot be adapted by means of this procedure.

Let us assume that the parallel training data have been partitioned into a set of `M` bilingual clusters, according to some criterion. On each cluster, language specific LMs are estimated, which are then organised into two language specific mixture models. All operations described so far are performed off-line. Now let us consider a source text or sentence to be translated. Before translation, the input is used to estimate optimal weights of the source language mixture through Expectation-Maximisation. This being done, the key step is to assume that such weights contain very valuable information about the distribution of the source language models, and this information can be passed on to the target mixture of language models by means

of a certain mapping of the source weights to the target weights. This mapping being done, the target language model mixture is then used as LM feature function by the SMT system. In the present work, such mapping will be performed by directly setting the target weights equal to the source weights. One could easily think of more sophisticated, and possibly more appropriate, ways of performing this mapping. However, this is a research direction that still needs to be explored.

In this chapter, two different frameworks for clustering the training data are considered. On the one hand, it will be first assumed that manually annotated texts are readily available, and the adaptation procedure will attempt to estimate the best weighting of these supervised clusters. On the other hand, since manual annotations are not always available, it will also be studied how to perform LM adaptation by means of unsupervised clustering, while still following the procedure described in Figure 3.1.

3.4 Supervised labelled data for language model adaptation

In this section, we describe how to take advantage of supervised information present in different bilingual corpora for the specific purpose of language model adaptation. By exploiting such labels, the bilingual corpus needed for training the SMT system will be divided into several different sub-corpora, and these sub-corpora will then be used within the adaptation framework presented in Figure 3.1. The sub-corpora built will serve as starting point for building adapted SMT systems as described in Section 3.3. For doing this, two different bilingual corpora will be considered: the IWSLT[a] and Nespole! (Lavie et al., 2006) corpora. For the purpose of differentiation, the term *cluster* will only be employed whenever these sub-corpora are built in a fully automated manner, whereas the term *sub-corpus* will be used in other cases.

Before pursuing with the description of the different labelled corpora employed and how these labels will be used, a brief overview is necessary so as to keep the motivation clear. Specifically, part of the IWSLT corpus contains translations of dialogues in a tourism domain, and has two kinds of labels:

- Labels grouping sentences into the dialogues where such sentences were originated. Since such dialogues are too short so as to estimate a LM, the dialogues will then be grouped into different clusters by means of an off-the-shelf clustering algorithm, treating each complete dialogue as a single sample for the purpose of clustering. These clusters (of dialogues) will then serve as starting point for the adaptation procedure described in Figure 3.1.

- Labels describing the nature of the speaker. Four different speaker types are considered, giving rise to four different sub-corpora, which, again, will serve as starting point for the adaptation procedure. No unsupervised clustering is applied in this case, since the labels themselves already divide the corpus into four sufficiently large sub-corpora.

On the other hand, the Nespole! corpus contains also dialogue translations in a tourism domain, and presents several labels:

[a] http://mastarpj.nict.go.jp/IWSLT2009/

- One label for each type of dialogue act. There are many types of labels, but the corpus will be divided into only three different sub-corpora so as to prevent sparsity: the two most different labels will constitute two of such sub-corpora, and the rest of the sentences will constitute the remaining sub-corpus.

Note, however, that the final experiments will be reported on the IWSLT corpus, which implies that the labels present in the Nespole! data first need to be carried on to the IWSLT data (the procedure for doing this is described below). No purely unsupervised clustering takes place here, although the sentences within the IWSLT corpus will be assigned different labels on the basis of likelihood.

With this overview in mind, a more detailed description of the corpora and methods used follows.

IWSLT

The corpus provided for the 2009 *International Workshop on Spoken Language Translation* (IWSLT) is composed of two different sub-corpora in Chinese–English: a larger corpus belonging to the general tourism domain and a smaller corpus also belonging to the tourism domain, but in the more specific context of hotel conversations. The larger sub-corpus, named *Basic Travel Expressions Corpus* (BTEC) has no manual annotations, whereas the smaller corpus, the *Challenge Task* (CT) corpus, does have manual annotations regarding speaker and dialogue number. Since the purpose is to perform adaptation, the experiments conducted focused on the CT data, which is the smaller part, in correct recognition results, Chinese–English (Zh→En) and English–Chinese (En→Zh) language pairs. The CT corpus includes for each sentence a dialogue identifier and the speaker class, i.e. agent, customer or interpreter. Table 3.1 reports statistics (running words and vocabulary size) of the training corpora used in our experiments after the preprocessing performed by means of the tools supplied by the organisers; the numbers for the two directions are different, despite the original texts are the same, because casing and punctuation have been removed from source texts, but kept on target texts. The reason for this is that the IWSLT campaign is about speech translation, and source texts are not provided as sentences as such, but in the form produced by the speech recogniser.

Zh→En	Chinese			English		
task	| W |	| V |	$\bar{s}$	| W |	| V |	$\bar{s}$
BTEC	148K	8408	7.4	183K	8344	9.1
CT	89K	3734	8.9	141K	3696	14.0

En→Zh	English			Chinese		
task	| W |	| V |	$\bar{s}$	| W |	| V |	$\bar{s}$
BTEC	153K	7294	7.7	172K	8428	8.6
CT	119K	3271	11.8	102K	3737	10.2

Table 3.1: Statistics of the IWSLT training data. |W| stands for running words, |V| for vocabulary size and $\bar{s}$ for average sentence length.

Since we are going to exploit speaker and dialogue annotations of the CT corpus, more detailed statistics are reported in Table 3.2. The figures regard the target side for the Zh→En task, being those for the other direction very similar.

	speaker	|W|	|V|	$\bar{s}$
agent	native	46.7K	2240	14.8
	interpreter	26.8K	1626	14.1
customer	native	33.3K	2082	13.9
	interpreter	33.8K	1878	12.9

Table 3.2: Speaker-based statistics of the CT training set.

By exploiting the annotation of the training and development texts, the data available can be subdivided in two different ways, one related to dialogues and the other to speakers. Then, these sub-corpora may be used for building different LMs, which will then, in turn, be considered for interpolation within Equation 3.2.

Dialogue based clustering: the CT data is split into 394 different dialogues representing a complete conversation between an agent and a customer. These dialogues are provided with identifiers, so that each single dialogue can be separated from the rest, and the different dialogues can be clustered as a whole, i.e. each one of the resulting clusters will contain several complete dialogues. For doing this, each dialogue was represented as a bag of both source and target words. The rationale behind this is to let the clustering algorithm decide which dialogues appear to be similar and are appropriate for building a specific LM. Since the clusters are formed relying on the words used in each dialogue, dialogues which have many words in common will end up in the same cluster, and dialogues which present less words in common will belong to different clusters, and (hopefully) topic-specific LMs will arise. For the clustering procedure, both source and target sides were used, as suggested by a slight performance gain observed in preliminary investigation. The number of clusters tested was 2, 4, 6 and 8, and on each of them a different LM was trained (see Figure 3.1). Additional LMs were built on the complete BTEC+CT data for smoothing purposes.

Speaker (agent/customer/interpreter) based grouping: In addition to the information described above, which identifies each dialogue as a whole, the CT data also contains information regarding the role of the current speaker. Since the CT data consists of interpreter-mediated conversations, four different roles appear: the customer, the agent, and the interpreter taking the role of either of the previous two. Hence, four different sub-corpora can be built exploiting this type of annotation, namely one of agent turns, one of customer turns, and two of interpreter turns which are translations of agent and customer utterances, respectively. Then, four different language models can be estimated on each side (i.e., language) of each sub-corpus. In this case, two additional LMs trained on the complete BTEC and BTEC+CT were included into the interpolation in Equation 3.2.

Nespole!

Nespole![b] (NEgotiating through SPOken Language in E-commerce) (Lavie et al., 2006) was a European Union funded project, running during years 2000-2002. It aimed at providing a system capable of supporting advanced needs in e-commerce and e-service by resorting to automatic speech-to-speech translation. In particular, one of the two implemented showcases supported multilingual negotiations and discussion between a tourist information/service provider (a so-called destination) and a customer who wanted to organise a trip exploring all available possibilities, including travel, accommodation, attractions and recreation, cultural events, dining and so on. Collected data mirrored such scenario. For the purposes of the work presented here, 58 Nespole! dialogues were used; they were collected in year 2000 involving Italian speakers, then translated into English and manually labelled in terms of dialogue acts. Table 3.3 reports corpus statistics regarding the English side of the dialogues, while Table 3.4 provides the (self-explanatory) labels and counters of the most frequent dialogue acts.

#turns	$\|$ W$\|$	$\|$V$\|$	$\bar{s}$
2522	15335	1344	6.1

Table 3.3: English side statistics of the Nespole! dialogues.

label	counter
`give-information`	963
`affirm`	408
`descriptive`	285
`request-information`	199
`acknowledge`	122
`greeting`	80
`negate-give-information`	62
`thank`	55
`request-action`	55
...	...
total	2522

Table 3.4: Most frequent Nespole! dialogue acts.

The English side of Nespole! data (see Section 3.7.1) was employed for subdividing the IWSLT training data. Since the Nespole! data includes labels regarding the kind of dialogue act of each utterance, the purpose was to carry on such information to the IWSLT training data, in order to mine possible differences in lexicon, syntactic structure or punctuation that different dialogue acts may entail. For doing this, the Nespole! corpus was first subdivided into three sub-corpora, according to the dialogue acts `give-information`, `request-information`, and all the rest. Then, three different 5-gram LMs were estimated on the English side of such sub-corpora. This being done, each English sentence of the

[b] http://nespole.itc.it

IWSLT corpus was labelled with the tags `give-information`, `request-information`, or `other`, according to which LM of the Nespole! sub-corpora assigns more probability to that specific sentence. Mirroring such assignments on the Chinese side of the IWSLT corpus gives rise to three different bilingual sub-corpora, and these three different sub-corpora can then be used as starting point for the adaptation procedure described in Figure 3.1. The rationale behind choosing `give-information`, `request-information` and `others` for the initial Nespole! subdivision is these first two dialogue acts are expected to label quite different sentences in terms of lexicon, syntactic structure and punctuation (when available).

Nespole! texts are quite different from IWSLT texts, although both of them are tourism-related. In this sense, it is specially illustrative that the cross-corpus perplexity is around 900, while the perplexity of IWSLT development/test sets ranges approximately from 50 to 200. Nevertheless, Nespole! data include valuable semantic annotation which might be worth exploiting. Note that, since the final evaluation experiments will be performed on the CT data and using the whole IWSLT corpus for training, the labels present in the CT data constitute reliable information towards building the final LM interpolation. In contrast, the information present in the Nespole! corpus first needs to be carried over to the IWSLT data.

3.5 Unsupervised clustering for language model adaptation

It should be clear that the fundamental intermediate step of the approach presented here is the clustering of bilingual training data. The elements of each cluster are sentences. Hence, the goal of this stage is to group together sentences which are similar to each other from the lexical point of view. However, since it is not always the case that supervised labels are readily available, in this section we explore the use of unsupervised clustering for this purpose. Unless differently specified, the clustering is performed by

- representing each sentence pair as a bag of both source and target words;
- setting the number of clusters to 4, since a preliminary investigation revealed this number as begin able to generate clusters quite specialised and not too sparse.

On both source and target sides, in addition to the 4 LMs trained on each cluster and for smoothing purposes, the LM built on the whole training data has also been considered.

In the following subsections, three different clustering schemes are described.

Direct clustering

As a first approach, we investigated clustering the training data directly.

Development-induced clustering

Although the direct clustering of the training data is the most straightforward choice, it might not be the best one, since by definition the goal of any adaptation procedure is to cover possible mismatches between training and development/test conditions. With this in mind, the idea is to cluster a given development set, and then attempt to mirror such clustering on

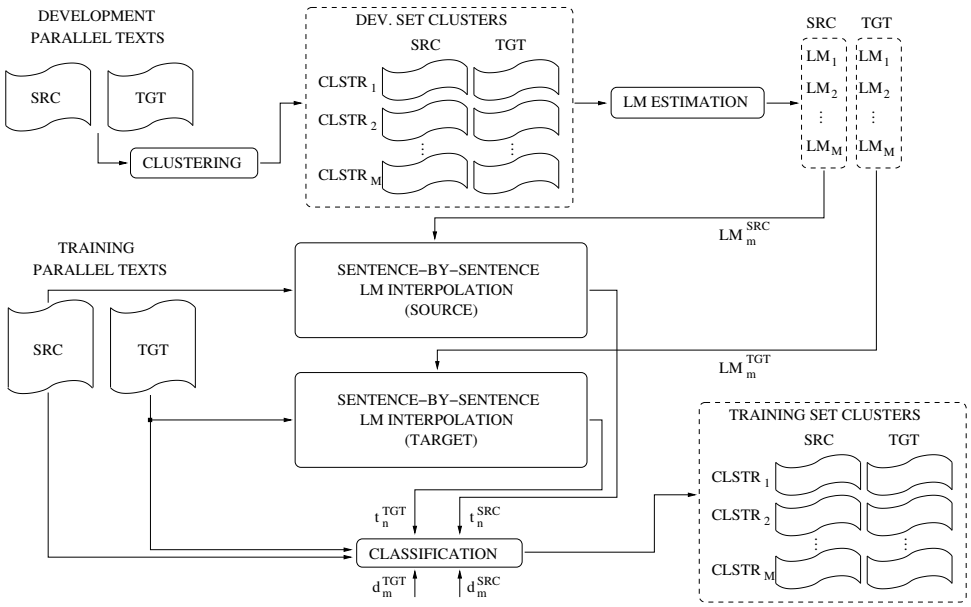


Figure 3.2: Procedure for obtaining development-induced clustering of the training data.

the training data. The procedure for doing this is shown in Figure 3.2 and is summarised in the following algorithm:

1. Cluster the bilingual development text
2. Estimate source and target LMs for each cluster from step (1)
3. Partition training data by classifying each sentence pair according to eq. 3.3 (see below)

In step (3), each bilingual training sentence n is assigned to the cluster $\hat{m}$ by the rule:

$$\hat{m} = \underset{m}{\text{argmax}}\, cos(\mathbf{t}_n^{\boldsymbol{x}}, \mathbf{d}_m^{\boldsymbol{x}}) + cos(\mathbf{t}_n^{\boldsymbol{y}}, \mathbf{d}_m^{\boldsymbol{y}}) \tag{3.3}$$

where $\mathbf{t}$ and $\mathbf{d}$ are vectors of M (the number of clusters) LM weights and the cosine between two vectors is defined as $cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{||\mathbf{a}||\,||\mathbf{b}||}$, with $\cdot$ being the dot product and $||\ ||$ being the 2-norm. In particular, $\mathbf{t}_n^{\boldsymbol{x}}$ is the set of LM weights that maximises the probability of the source sentence n of the training text, according to the linear interpolation of source LMs estimated in step (2). $\mathbf{t}_n^{\boldsymbol{y}}$ is the twin of $\mathbf{t}_n^{\boldsymbol{x}}$ for the target side. $\mathbf{d}_m^{\boldsymbol{x}}$ is the vector of weights which maximise the probability of again the source LMs of step (2) but on the whole source side of cluster m of the development set. $\mathbf{d}_m^{\boldsymbol{y}}$ is the twin of $\mathbf{d}_m^{\boldsymbol{x}}$ for the target side.

The intuitive explanation of eq. 3.3 relies on the meaning of components of vectors $\mathbf{t}$ and $\mathbf{d}$. Let us start by the fact that in some sense a LM trained from a specific cluster is a compact representation of the sentences in that cluster; hence, the optimisation of LM weights on a text

provides, through each single weight, a measure of the similarity of that text with a specific LM, that is a specific cluster. Vectors $\mathbf{t}$ and $\mathbf{d}$ can then be considered as "fingerprints" of each training sentence and development cluster, respectively. The $cos()$ operation on them is then applied to compute the similarity of training sentences with each cluster m.

Test-induced clustering

For inducing the clustering of the bitext training data it is possible to use the test set instead of the development set. Since in this case the target side is not available, the clustering is performed only on the source data, and the classification rule of eq. 3.3 is modified accordingly:

$$\hat{m} = \underset{m}{\text{argmax}}\, cos(\mathbf{t}_n^{\boldsymbol{x}}, \mathbf{d}_m^{\boldsymbol{x}}) \tag{3.4}$$

In this case, $\mathbf{d}_m^{\boldsymbol{x}}$ refers to the vector of weights which maximise the probability of source LMs on the source side of cluster m of the partitioning of the *test* set. Note that even if eq. 3.4 relies only on the source side, it is used to classify both sides of each sentence n of the training data.

The idea behind performing a test-induced clustering is that of taking profit of the information available in the actual text to be translated, with the purpose of grouping together test sentences which are similar. Nevertheless, the possible benefits of using such information may not be completely reliable, since only the source side is available and the clustering is instead induced on bilingual data. Note, however, that for performing this kind of clustering the test data must be known beforehand.

3.6 Weight optimisation strategies

Once different clusters have been obtained and appropriate LMs have been estimated for each set of clusters, a set of weights is needed for performing the actual interpolation of LMs that will be used in translation time. For this purpose, three different approaches were investigated, each one with a different degree of granularity.

Set specific weights

The LM-interpolation weights were estimated on the source side of the complete test set. This approach, which is the most straightforward, has nevertheless an important drawback: the estimated weights are those that well model the whole test set on average, without considering possibly significant differences between specific sentences. Hence, the potential benefit of estimating several LMs may fade.

Sentence specific weights

In this case, one specific set of weights is estimated for each sentence of the test set. By doing so, the purpose is to allow complete freedom to the EM procedure when assigning the LM weights, and hence achieve better results when separating the training corpus into several subsets. However, weights computed in such a manner may be less reliable, since the estimation is performed on few data (one single sentence).

Two-step weight estimation

This approach merges the previous two in the attempt of keeping their advantages and overcoming the drawbacks. Once sentence specific weights have been computed, each (source) sentence is assigned to the specific cluster corresponding to the most weighted LM. This being done, one set of weights can be re-estimated for each one of the clusters obtained in this way. This approach has the intuitive benefit of mirroring the clustering of the training data into the test set, while still avoiding the possible data sparseness issue that can affect the sentence specific weight estimation. This procedure is illustrated in Figure 3.3.

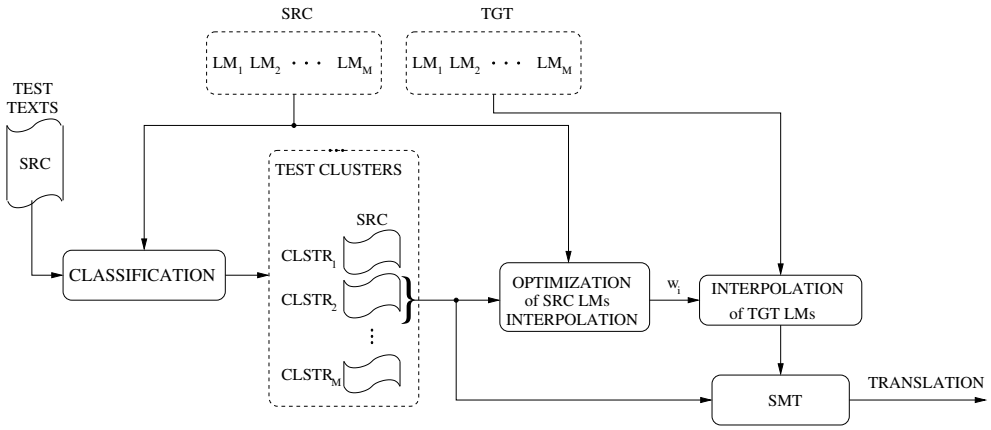


Figure 3.3: Two-step weight estimation technique.

3.7 Experimental results

This section reports the results of both language model adaptation strategies described in Sections 3.4 and 3.5. Translation quality results will be reported in terms of BLEU and TER. However, with the purpose of getting some insight about what is really happening during the adaptation process, additional results will be reported in terms of perplexity (PP). Perplexity (Bahl et al., 1983) is a measure stemming from information theory, and is defined as 2 raised to the power of the entropy of a given test set $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m, \ldots, \boldsymbol{y}_M\}$, such that

$$PP(\boldsymbol{Y}) = 2^{\frac{1}{N}\sum_m \log_2 p_{\mathrm{LM}}(\boldsymbol{y}_m)}, \tag{3.5}$$

for a given language model LM and where N stands for the total number of words in the test set. In more intuitive terms, perplexity is often understood as the average number of possible words that are likely to follow a given prefix. However, perplexity may be used for two different (but complementary) purposes: on the one hand, perplexity may be used to compare two different language models, and on the other hand it may also be used to assess the complexity of a given task. In this chapter, perplexity will be used with the purpose of comparing different language models, i.e. the monolithic baseline language model with the

interpolated language model built by means of the clustering techniques described. However, it must be noted that improvements in perplexity are not always mirrored by improvements in system performance. This implies that perplexity may help towards establishing which language model performs best, but such conclusion must always be backed up by coherent results in terms of system performance – translation quality in the case of SMT.

Whenever unsupervised clustering is required, such as in the case of exploiting dialogue annotation (Section 3.4) or in the case of building unsupervised clusters (Section 3.5), such clustering will be performed by means of the CLUTO[c] package. Its default setup includes the `direct` clustering algorithm, which computes the *k*-way clustering directly by means of the K-means algorithm (Zhao and Karypis, 2005). The cosine distance was used as criterion function.

3.7.1 Experiments using supervised labels

In order to study the similarity, or better the differences, between training and testing conditions, the statistics shown in Table 3.2 for the CT training data were also computed for the development data set (Table 3.5). It clearly results that the two sets differ not only in sentence length, but also in terms of distribution of utterances from the interpreter. We will see later if and in which cases this mismatch affects system performance.

	speaker	|W|	|V|	$\bar{s}$
agent	native	2.5K	427	15.1
	interpreter	0.8K	218	13.2
customer	native	0.5K	152	11.8
	interpreter	1.7K	307	12.3

Table 3.5: Speaker-based statistics of the CT development set.

So as to provide an upper bound of the performance that can be reached with the supervised adaptation technique presented in this chapter, optimal sentence specific weights have also been estimated on the reference translations.

Coherently to what has been written at the beginning of this section, experiments were performed on the development sets of the Challenge Task of IWSLT09, Zh→En/En→Zh, correct recognition result transcripts tasks. They were split in two parts (DEV1 including 4 dialogues, DEV2 with 6 dialogues) which were alternatively used for MERT and evaluation.

Results are provided in Figures 3.4-3.11. Each of them includes two plots: the plot on the top shows BLEU scores, the one on the bottom displays perplexity. Figures 3.4, 3.5, 3.8 and 3.9 report results obtained by dynamically estimating the interpolation weights at the sentence level (Section 3.6), while Figures 3.6, 3.7, 3.10 and 3.11 refer to the two-step technique (Section 3.6). Finally, Figures 3.4, 3.5, 3.6 and 3.7 show performance for the En→Zh direction, while Figures 3.8, 3.9, 3.10 and 3.11 for the Zh→En task.

The five curves in each plot refer to different systems:

`baseline`: SMT system using one single LM estimated on the whole training corpus;

[c] Available from http://glaros.dtc.umn.edu/gkhome/views/cluto

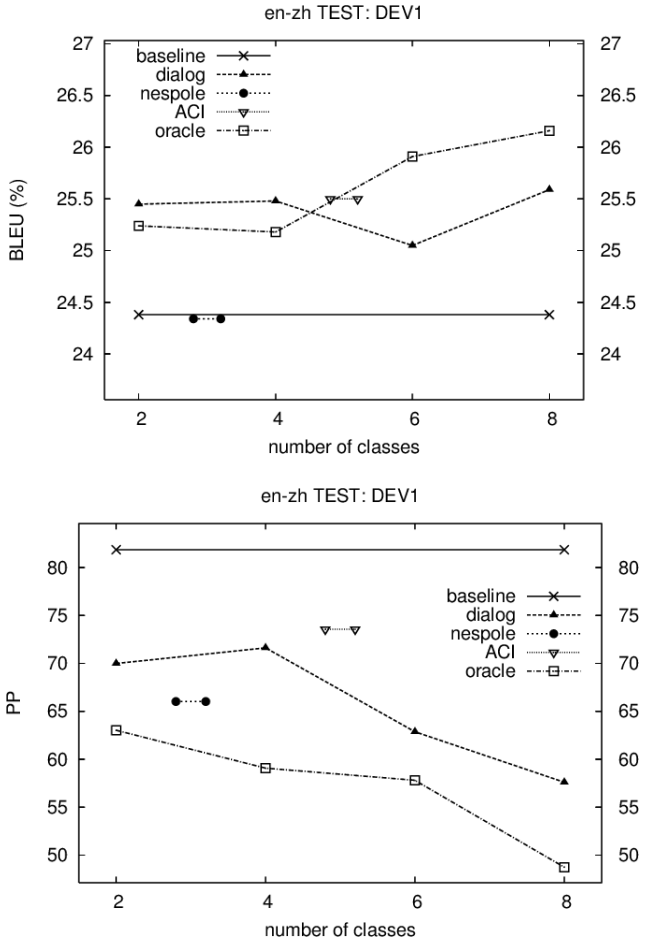


Figure 3.4: En→Zh results (BLEU scores and perplexity) for set DEV1 with different grouping methods, sentence specific weight estimation.

`dialogue`: interpolation of LMs built on the dialogue based clustering as described in Section 3.4;

`nespole`: interpolation of LMs built on the sub-corpora induced by the Nespole! data (Section 3.4);

`ACI`: interpolation of LMs built on the speaker-based sub-corpora as described in Section 3.4;

`oracle`: the LMs are those built on the dialogue basis, but the interpolation weights are estimated by means of an oracle. So as to provide an upper bound of the performance that can be reached with the adaptation technique presented in Section 3.4, optimal

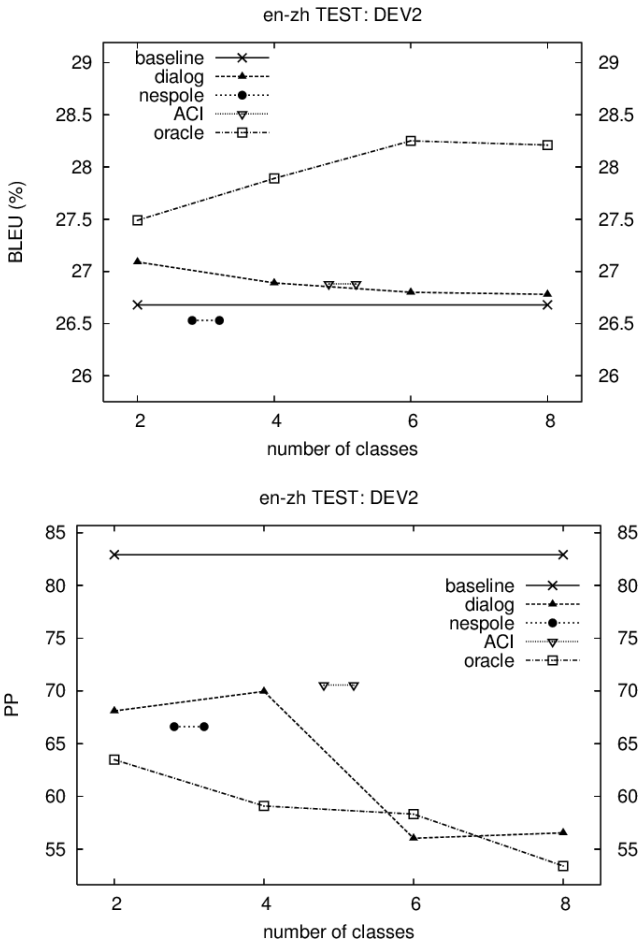


Figure 3.5: En→Zh results (BLEU scores and perplexity) for set DEV2 with different grouping methods, sentence specific weight estimation.

sentence specific weights have also been estimated on the reference translations.

In the case of the `nespole` and `ACI` curves, the number of classes is fixed to 3 and 5, respectively, and should be hence plotted as a single point, but is shown in the plots as a short segment for the purpose of visibility.

Results achieved by interpolating LMs with weights estimated at the test set level (Section 3.6) are not reported for the sake of simplicity and because they are not better than those of the competing techniques, as expected.

Before the detailed analysis, a general comment is that in terms of perplexity the idea of building LMs on some motivated partition of the training data and then interpolate them with weights estimated on the actual input performs very well, yielding significant improvements whatever the grouping technique, the number of sub-corpora (LMs) and the scheme followed

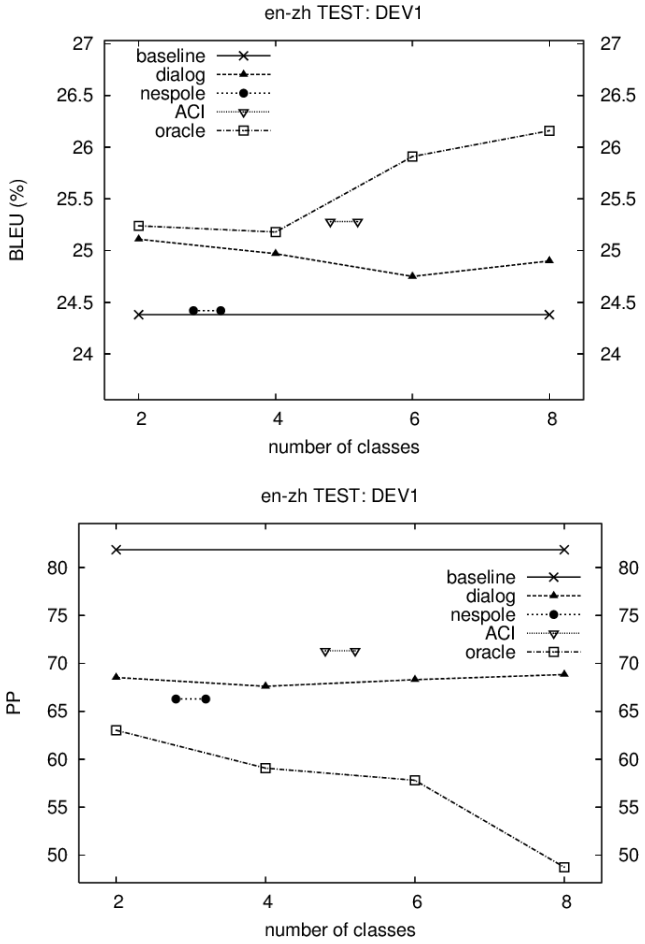


Figure 3.6: En→Zh results (BLEU scores and perplexity) for set DEV1 with different grouping methods, two-step weight estimation.

for the estimation of interpolation weights. Moreover, the BLEU score of the `oracle` system confirms that the approach is really appealing. On the other side, for the fair systems the impressive improvement in terms of perplexity is not always mirrored in the BLEU score, especially for sub-corpora built exploiting either Nespole! annotation or speaker information, for which even a degradation is observed in some cases.

In relation to the experimental outcomes, the following additional remarks can be made:

- the `oracle` curves are uni-modal and mostly present a peak at six clusters, which is then the optimal number of LMs to be interpolated;
- the shape of the curves of the two-step procedure (Figures 3.6, 3.7, 3.10 and 3.11), although are not higher than those of the estimation performed on single sentences

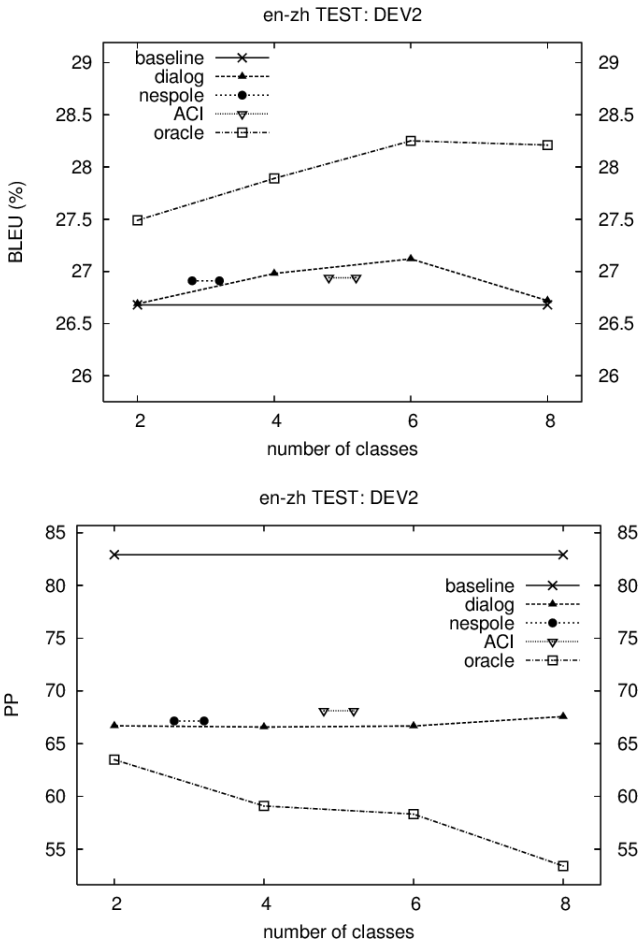


Figure 3.7: En→Zh results (BLEU scores and perplexity) for set DEV2 with different grouping methods, two-step weight estimation.

(Figures 3.4,3.5,3.8,3.9), are more similar to those of the oracle (uni-modal), fact that makes its behaviour more predictable;

- the `dialogue` based clustering improves or at least does not worsen too much baseline BLEU scores, even if it tends to be quite far from the oracle quality; there is no clear evidence about the optimal number of clusters;
- `ACI` works quite well for the En→Zh task but not for the Zh→En direction;
- `nespole` partitioning does not seem to be effective in terms of BLEU score;
- performance by switching the role of DEV1 and DEV2 is quite different;

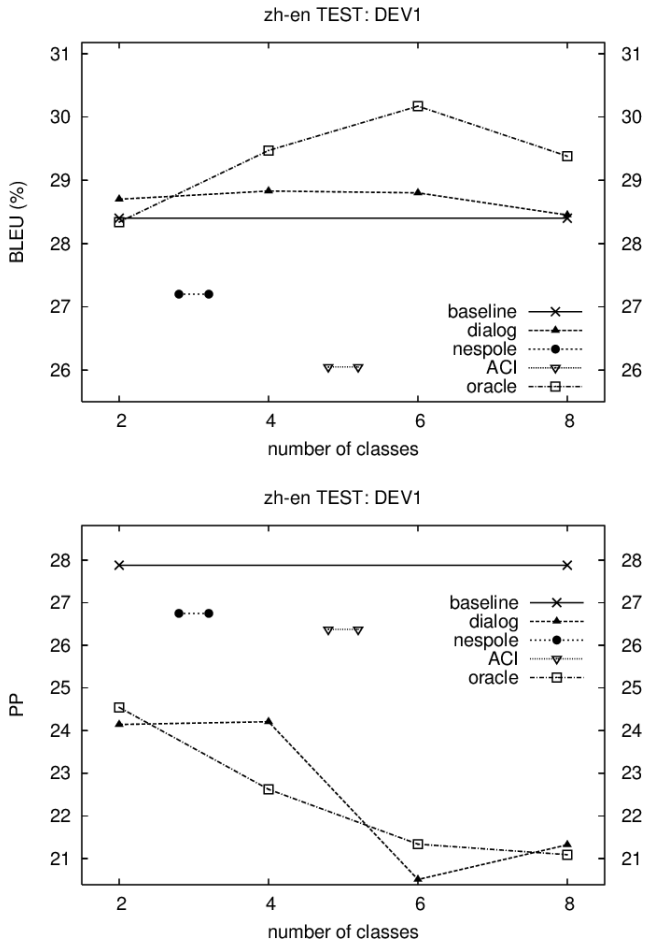


Figure 3.8: Zh→En results (BLEU scores and perplexity) for set DEV1 with different grouping methods, sentence specific weight estimation.

- improvements over the baseline are larger on En→Zh direction than on Zh→En.

It is important to stress the fact that training/development and test conditions were quite different in the experiments conducted. This was already pointed out by the comparison of figures in Tables 3.2 and 3.5, but it is even more evident by observing that MERT is effective only for the En→Zh direction and when DEV2 and DEV1 are used for development and evaluation respectively, while it degrades the performance of the initial setup in all the other three cases; Table 3.6 gathers the variations of the BLEU score between initial and final configurations of the SMT system for the two directions (Zh→En and En→Zh) and with the two possible roles for DEV1 and DEV2. This disappointing behaviour is probably due to the too small size of DEV1, fact that could also explain why our adaptation technique does not work very well on DEV2, i.e. when DEV1 is used for development.

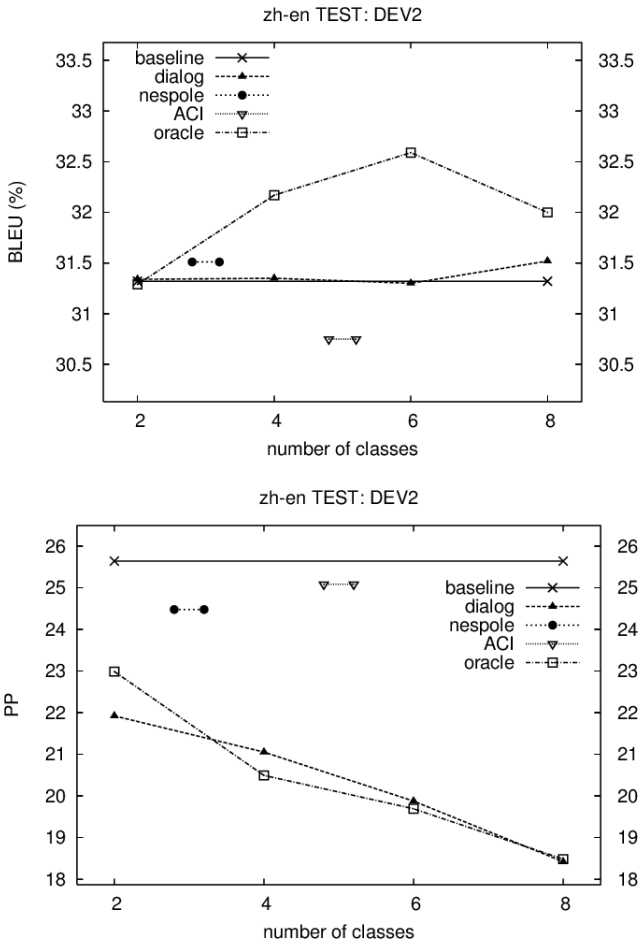


Figure 3.9: Zh→En results (BLEU scores and perplexity) for set DEV2 with different grouping methods, sentence specific weight estimation.

test on	mert on	Δ BLEU CE	Δ BLEU EC
DEV1	DEV2	-0.19	+3.39
DEV2	DEV1	-0.67	-1.12

Table 3.6: MERT effect on the BLEU score.

It can also be observed that, in some rare cases, that `oracle` BLEU scores drop below the `dialogue` scores. This could be due to the fact that we assume that the LM interpolation weights computed on the reference sentence are the ones that best exploit the provided models. However, such assumption could not be true in the case of a severe mismatch between

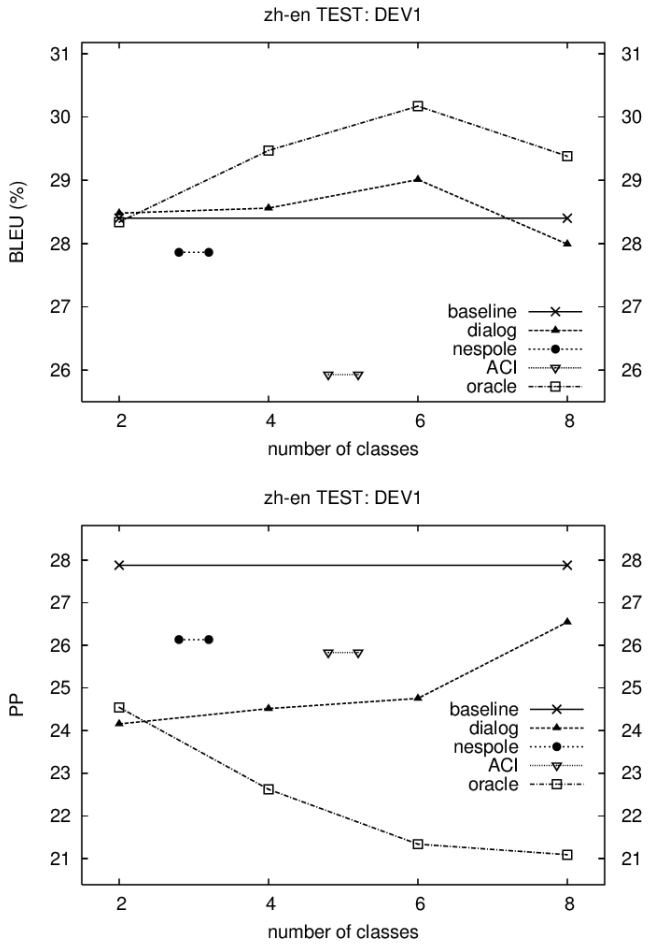


Figure 3.10: Zh→En results (BLEU scores and perplexity) for set DEV1 with different grouping methods, two-step weight estimation.

such models and reference sentences, leading to the possibility of achieving better scores with other weights.

A final remark is needed on the fluctuating performance of the ACI sub-corpora. Its purpose is to obtain speaker-role specific LMs, which should theoretically perform better than generic LMs when it is possible to know which is the role of the actual speaker. However, if training and test conditions within each dialogue role present a severe mismatch, as seems to be the case according to Tables 3.2 and 3.5, such an approach is bound to yield a very limited benefit, if any.

Despite all the precautions required by the fact that the experimental outcomes are not unquestionable, an encouraging conclusion can be drawn. It emerges that the LM adaptation approach proposed here is promising and can guarantee quite stable improvements over the

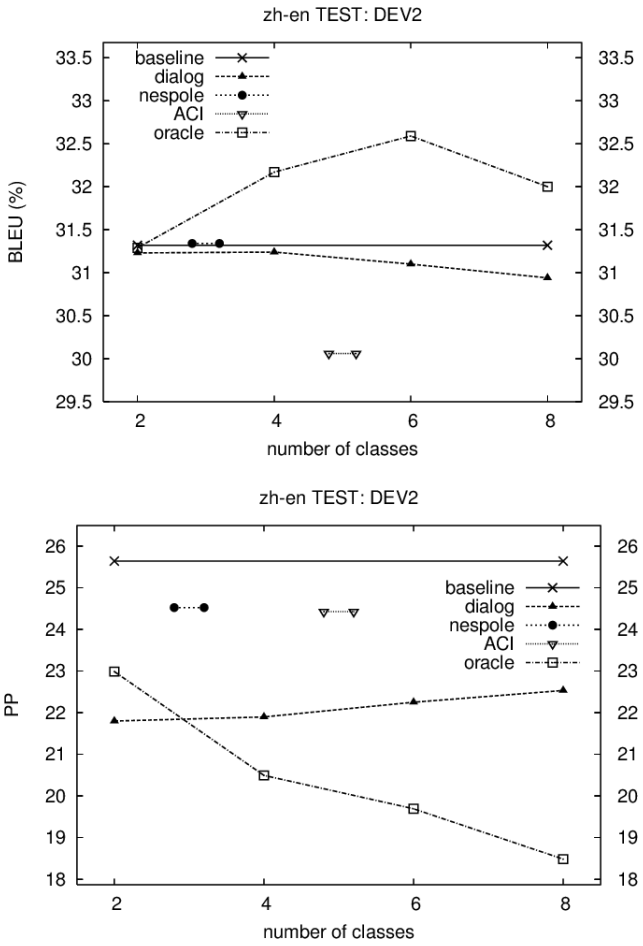


Figure 3.11: Zh→En results (BLEU scores and perplexity) for set DEV2 with different grouping methods, two-step weight estimation.

baseline quality when the clustering is built at the level of dialogues and the interpolation weights are estimated with the two-step scheme.

3.7.2 Unsupervised clustering experiments

The experiments conducted for assessing the unsupervised clustering LM adaptation technique were performed on the Europarl corpus, in the partition established for the WMT06 workshop (see Section 1.4). In this case, the languages involved in the experimentation were English→German, English→Spanish and English→French. The baseline and subsequent systems were built by means of the Moses SMT toolkit, and the weights λ of the log-linear model were optimised by means of MERT for the baseline system on the `Devel.` set, and

Language pair	Weight optimisation	PP	BLEU	TER	Signif BLEU/TER
En–Es	baseline	78.5	30.8	54.9	–
	sentence	71.3	30.4	54.6	yes/yes
	two-steps	71.2	30.3	54.5	yes/yes
	test set	100.1	30.3	54.5	yes/yes
En–De	baseline	141.5	19.0	67.4	–
	sentence	129.0	18.2	67.4	yes/no
	two-steps	129.7	18.1	67.4	yes/no
	test set	202.3	18.0	67.6	yes/no
En–Fr	baseline	50.0	32.9	55.3	–
	sentence	45.4	32.7	55.0	no/yes
	two-steps	45.5	32.6	54.9	yes/yes
	test set	64.5	32.5	55.0	yes/yes

Table 3.7: Performance of the direct clustering approach.

then re-used for all other systems. Although there could be reasons for re-running MERT when the LM changes, this was done so in order to better isolate the effects of including different LMs into the SMT system. As baseline LM, a 5-gram word-based LM was estimated on the target side of the training corpus, smoothed according to the improved Kneser-Ney technique (Chen and Goodman, 1999), by means of the SRILM (Stolcke, 2002) toolkit. The final translation quality was measured on the `Devtest` set.

The adaptation procedures presented in Section 3.5 have been experimentally assessed by translating different test sets, whose quality was measured in terms of BLEU (Papineni et al., 2001) and TER (Snover et al., 2006). *Pairwise* statistical significance tests using paired bootstrap re-sampling (see Section 1.2.2) were also computed with ten thousand bootstrap repetitions. These tests, showing whether the improvement (or drop) in translation quality with respect to the baseline performance is significant at 95% confidence level, were computed for both BLEU and TER and are provided in the `Signif` column. Note, however, that even though paired bootstrap re-sampling proves some systems to be statistically differentiable, confidence intervals were in most of the cases in the range of 0.7, both in case of TER and in case of BLEU.

Finally, the column `PP` shows the perplexity value of either the single LM (baseline) or the interpolation of LMs (other cases) computed on the test set references.

Direct clustering

Results observed by directly clustering the training data are shown in Table 3.7, for all three weight optimisation schemes and for all three language pairs.

A degradation of the BLEU score is observed in any condition, while TER slightly improves for the En–Es and En–Fr pairs, especially when either the sentence-based or the two-steps estimation schemes are adopted. However, since results are not coherent for both scores, it cannot be definitely stated whether this form of LM adaptation overcomes the use of the single baseline LM.

Language pair	Weight optimisation	PP	BLEU	TER	Signif BLEU/TER
En–Es	baseline	78.5	30.8	54.9	–
	sentence	68.3	**31.3**	54.4	yes/yes
	two-steps	68.3	**31.3**	**54.3**	yes/yes
	test set	105.6	30.9	54.6	yes/yes
En–De	baseline	141.5	19.0	67.4	–
	Sentence	126.0	**19.2**	**66.7**	yes/yes
	two-steps	126.3	**19.2**	**66.7**	yes/yes
	test set	206.6	18.7	67.2	yes/no
En–Fr	baseline	50.0	32.9	55.3	–
	sentence	43.5	33.2	54.9	yes/yes
	two-steps	43.5	**33.3**	**54.8**	yes/yes
	test set	65.0	32.9	55.1	no/yes

Table 3.8: Performance of the development-induced clustering approach. The best results are marked in bold.

Development-induced clustering

Results for the development-induced clustering are reported in Table 3.8. In this case, the LM adaptation does improve the baseline consistently, for both scores and significantly in almost every setup. Again, the best performing weight optimisation scheme is the two-steps one, which improves the baseline in all language pairs in a statistically significant way. Performances comparable to those of two-steps optimisation are obtained also with weights estimated at the single test sentence level. Again, the optimisation of weights on the whole test set does not seem to be appropriate.

Test-induced clustering

Lastly, Table 3.9 collects results when the clustering of training data is induced by the test set. This kind of clustering seems not to be able to exploit the test information provided to the system; in fact, BLEU is non-differentiable from the baseline in almost every setup, while TER is improved only at a limited extent. Concerning the weight optimisation, here the best choice is to perform it on the whole test set, differently from what happened in the other types of clustering. This could be originated from the fact that LMs are built on clusters induced by just the test set. For this reason, in this specific case the use of the whole test set allows an effective trade-off between the estimation of weights which are good on average on the whole test set and the sparseness of data on which the optimisation is done. Nevertheless, it is worth noticing that differences in translation quality are mostly not statistically significant.

Results in Tables 3.7, 3.8 and 3.9 show the different impact that the proposed clustering and weight optimisation schemes for LM adaptation have on MT performance. In particular, the best scores measured in our experiments, marked in bold in Table 3.8, are achieved when using development-induced clustering combined with the two-steps (or sentence-based) weight optimisation. With this setup, the translation quality always improves the one obtained

Language pair	Weight optimisation	PP	BLEU	TER	Signif BLEU/TER
En–Es	baseline	78.5	30.8	54.9	–
	sentence	72.4	30.9	54.6	no/yes
	two-steps	72.2	30.9	54.6	no/yes
	test set	105.7	31.0	54.6	yes/yes
En–De	baseline	141.5	19.0	67.4	–
	sentence	133.7	18.9	67.3	no/no
	two-steps	133.9	18.9	67.3	no/no
	test set	204.4	18.9	67.1	no/yes
En–Fr	baseline	50.0	32.9	55.3	–
	sentence	46.6	32.8	55.2	no/no
	two-steps	46.4	32.8	55.3	no/no
	test set	65.2	33.0	55.2	no/no

Table 3.9: Performance of the test-induced clustering approach.

by the baseline system. Such results, which are statistically significant and coherent throughout all language pairs and for both considered evaluation scores, prove that there is a potential benefit behind the use of n-gram mixtures in SMT, also in the non-supervised setup.

From another viewpoint, it seems that the sentence-based interpolation technique is able to yield the same translation quality than the two-steps weight optimisation. This should indirectly prove that the input sentence alone contains sufficient information to make the interpolation procedure stable enough. In fact, average sentence length for the test sets ranges from 33 words per sentence for French, to 27 words per sentence for German, i.e. fairly long sentences. Given this experimental evidence and the fact that it is computationally cheaper, the sentence-based optimisation should be the first choice in presence of quite long input sentences.

It must also be noted that, although all the subsets of the Europarl corpora belong to the same domain, they were not extracted randomly: specifically, the training corpus comprises data from year 1997 to year 2003, although the development and test data are extracted from the fourth quarter of year 2000. This fact should explain the good results obtained with the development-induced clustering, since both test and development sets belong to a very narrow time frame, in which the topics being debated in the European Parliament were likely similar. Hence, development-induced clustering may be able to make a better use of the uneven distribution of training and development/test data, since it resembles the test data, and contains bilingual information (as opposed to test-induced clustering).

The fact that test-driven clustering only relies on source-sentence information is an important drawback that cannot be ignored: preliminary investigations revealed that including both source and target information into the clustering procedure did have an important impact, which is evidenced in this case as well. Although it might seem that monolingual clustering relies on half of the information of bilingual clustering, this is even optimistic: in fact, bilingual clustering does not only take into account both source and target sides, but also the interaction between the two, since it also takes into account whether a given source word

cooccurs with a given target word.

3.8 Conclusions and future work

In this chapter, a technique for adapting the LM of SMT systems to the actual input has been presented. The assumption is that the LM is provided as a linear interpolation of sub-LMs, each estimated on a specific portion of the training data. The interpolation weights are then estimated dynamically on the text to be translated via a maximum likelihood EM-based procedure.

Different methods for subdividing the training data have been presented, both in a supervised and in an unsupervised manner. Regarding supervised subdivision, manually annotated texts have been used for subdividing the training data; regarding unsupervised clustering, different strategies were presented, some of them attempting to take advantage of development or test information.

Different schemes for estimating the interpolation weights have also been experimentally tested when combined with both supervised and unsupervised clustering strategies.

Results have shown that small improvements may be obtained by partitioning the training data into more specific sub-corpora, and learning independent language models from them. However, these improvements were not always statistically significant. In the case of unsupervised clustering, the best results were achieved by clustering the training data by exploiting both sides (source and target) of the development set, and estimating the weights at the sentence level or by means of the two-step approach.

Results achieved in this work reveal that the improvements that can be obtained by our LM adaptation approach greatly depend on the subdivision technique employed. Since here only the surface form of single words has been used for clustering the training data, possible alternatives include clustering the training data according to n-gram or PoS-tag information.

Another issue which deserves an investigation regards the interpolation of target LMs by re-using weights estimated for the optimal interpolation of source LMs. In fact, although it appears as a reasonable choice, it could happen that the likelihood on the target side is maximised with different weights than those which ensures the maximum likelihood on the source side. A source-to-target weight map could be learnt from a parallel development/training set.

Lastly, future work also involves comparing the language model adaptation technique presented here with other thechniques present in the literature, such as the ones described in Section 3.2. However, it is also worth noting that the technique presented here is compatible with the most of the techniques described in the above-mentioned section, and hence should not be viewed as competing approaches.

The biggest part of the work done in this chapter was done during an internship at the *Fondazione Bruno Kessler*, in collaboration with M. Federico and M. Cettolo. The first publication about the supervised LM adaptation technique was published in an international workshop:

- **G. Sanchis-Trilles**, M. Cettolo, N. Bertoldi and M. Federico Online Language Model Adaptation for Spoken Dialog Translation. In *Proceedings of the International Workshop on Spoken Language Translation, IWSLT 2009*, pages 160–167, Tokyo (Japan), December 2009.

The work presented above was also used within an SMT system submitted for evaluation to that same workshop. Notably, the English↔Chinese systems submitted ranked second and third, depending on the task and evaluation method.

- N. Bertoldi, A. Bisazza, M. Cettolo, **G. Sanchis-Trilles** and M. Federico FBK @ IWSLT 2009. In *Proceedings of the International Workshop on Spoken Language Translation, IWSLT 2009*, pages 160–167, Tokyo (Japan), December 2009.

The work about unsupervised LM adaptation was presented in an international conference:

- **G. Sanchis-Trilles** and M. Cettolo Online Language Model Adaptation via N-gram Mixtures for Statistical Machine Translation. In *Proceedings of the 14th Conference of the European Association for Machine Translation, EAMT 2010,* Saint-Raphaël, (France), May 2010.

In addition, work on bilingual sentence clustering derived from the work presented in this chapter was presented in another international workshop:

- J. Andrés-Ferrer, **G. Sanchis-Trilles** and F. Casacuberta Similarity Word-Sequence Kernels for Sentence Clustering. In *Proceedings of the 8th International Workshop on Statistical Pattern Recognition, S+SSPR 2010,* Cesme (Turkey), August 2010.

Bibliography

Lalit R. Bahl, Frederik Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, 1983.

Jerome R. Bellagarda. An overview of statistical language model adaptation. In *Proceedings of ISCA Workshop on Adaptation Methods for Speech Recognition*, pages 165–174, Sophia-Antipolis, France, 2001.

Nicola Bertoldi and Marcello Federico. Domain adaptation in statistical machine translation with monolingual resources. In *Proceedings of the EACL fourth workshop on Statistical Machine Translation*, pages 182–189, March 30–31 2009.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 4(13):359–393, 1999.

Jorge Civera and Alfons Juan. Domain adaptation in statistical machine translation with mixture modelling. In *Proceedings of the ACL second workhop on Statistical Machine Translation*, pages 177–180, June 23 2007.

Philipp Koehn and Josh Schroeder. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the ACL second workhop on Statistical Machine Translation*, pages 224–227, June 23 2007.

Roland Kuhn and Renato De Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6): 570–583, 1990.

Antonio Lagarda and Alfons Juan. Topic detection and classification techniques. In *WP4 deliverable*, TransType2, 2003.

Alon Lavie, Fabio Pianesi, and Lori S. Levin. The NESPOLE! System for multilingual speech communication over the Internet. *IEEE Transactions on Audio, Speech & Language Processing*, 14(5):1664–1673, 2006.

Yajuan Lü, Jin Huang, and Qun Liu. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings the conference on Empirical Methods in Natural Language Processing and Computational Natural Language Processing*, pages 343–350, June 28–30 2007.

Renato De Mori and Marcello Federico. Language model adaptation. In K. Pointing, editor, *New Advances and Trends in Speech Recognition and Coding*, NATO ASI Series F, pages 280–301. Springer Verlag, Germany, 1999.

Laurent Nepveu, Guy Lapalme, Philippe Langlais, and George Foster. Adaptive language and translation models for interactive machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 190–197, July 25–26 2004.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Technical Report RC22176 (W0109-022)*, 2001.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of conference of the Association for Machine Translation in the Americas*, pages 223–231, August 8–12 2006.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of the 7th international conference on Spoken Language Processing, 2002*, pages 901–904, September 16–20 2002.

Hirofumi Yamamoto and Eiichiro Sumita. Bilingual cluster based models for statistical machine translation. In *Proceedings the conference on Empirical Methods in Natural Language Processing and Computational Natural Language Processing*, pages 514–523, 2007.

Bing Zhao, Matthias Eck, and Stephan Vogel. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the international conference on Computational Linguistics*, pages 411–417, August 23–27 2004.

Ying Zhao and George Karypis. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, 2005.

CHAPTER 4

Bayesian translation model adaptation

Los aviones vuelan y no mueven las alas.

Francisco Casacuberta

Contents

4.1 Introduction **85**
4.2 Related work **86**
4.3 Bayesian predictive adaptation for SMT **88**
4.4 Online Bayesian adaptation **94**
4.5 Bayesian adaptation for model stabilisation **95**
4.6 Sampling methods **96**
4.7 Practical approximations **100**
4.8 Experiments **102**
4.9 Conclusions and future work **119**
Bibliography **123**

"What good is your reality, when justice fails and dishonesty is glossed over and the ones who keep faith suffer. Helene kept her bargain about Ellis and so did I. What good is your reality then?"

"Look here," Furii said. "I never promised you a rose garden. I never promised you perfect justice..." (She remembered Tilda suddenly, breaking out of the hospital in Nurenburg, disappearing into the swastika-city, and coming back laughing that hard, rasping parody of laughter. "Sholom Aleichem, Doctor, they are crazier than I am!")... and I never promised you peace or happiness. My help is so that you can be free to fight for all of these things. The only reality I offer is challenge, and being well is being free to accept it or not at whatever level you are capable. I never promise lies, and the rose-garden world of perfection is a lie... and a bore, too!"

I never promised you a rose garden. Joanne Greenberg.

"Lo bueno es tu realidad, cuando la justicia y la deshonestidad no es pasado por alto y los que mantienen la fe sufre. Helen mantiene su negocio de Ellis y yo también ¿Qué, pues bueno es su realidad?"

"Mira aquí", dijo Furii. "Nunca te prometí un jardín de rosas. Yo nunca te prometí ..." justicia perfecta (De pronto recordó Tilda, saliendo del hospital en Nurenburg, desapareciendo en la cruz gamada de la ciudad, y volver riendo que la parodia dura, áspera de la risa. "Sholem Aleijem, el doctor, son más locos que yo!")... Y yo nunca te prometí la paz o la felicidad. Mi ayuda es para que pueda ser libre de luchar por todas estas cosas. La única realidad que ofrecen es el desafío, y de ser así se está libre de aceptar o no al nivel que sea que usted es capaz. Prometo nunca miente, y el mundo jardín de rosas de la perfección es una mentira... y un taladro, también!"

Yo nunca te prometí un jardín de rosas. Google Translate.

4.1 Introduction

Nowadays, there are large amounts of bilingual data available for very specific domains, such as parliamentary speeches or news-wire articles. However, typical IMT systems are usually used for aiding human translators in very different tasks, such as translating patient information leaflets or printer manuals. Such situation leads to a strong discrepancy between the data on which the underlying SMT system has been trained and the data on which it is going to be applied. In order to bridge this discrepancy, *model adaptation* techniques are often used. The aim of such techniques is to make the best use of a small amount of adaptation data, belonging to the domain which is going to be translated, in order to take profit of the generality provided by the massive amount of data available in more resourceful domains.

Adaptation has become a very popular issue in machine translation (Koehn, 2010). Typically, the adaptation problem arises when two very different sets of training data are available, which implies also that two different sets of model parameters can be obtained. The first set of data, which will be referred to as training data $\mathcal{T}$, is typically very large and usually rather generic in domain. The second set of data, referred to as adaptation data $\mathcal{A}$, is usually overwhelmingly smaller than $\mathcal{T}$, but belongs the specific task of interest. In such scenario, the challenge is to modify the output of our system appropriately by taking into consideration both $\mathcal{T}$ and $\mathcal{A}$: on the one hand, making use of $\mathcal{T}$ is ought to provide robustness to the estimation of the model parameters $\boldsymbol{\theta}$, and on the other hand $\mathcal{A}$ should introduce a certain bias towards the specific task that is being tackled. This definition of adaptation is specially appropriate for the Bayesian learning paradigm, where the model parameters $\boldsymbol{\theta}$ are treated as (hidden) random variables which are governed by some kind of a priori distribution $p(\boldsymbol{\theta})$. This distribution represents our prior knowledge about what values for $\boldsymbol{\theta}$ are good estimates. Estimating $p(\boldsymbol{\theta})$ by using $\mathcal{T}$, and considering $\mathcal{A}$ within the Bayesian predictive distribution to be used when translating a given sentence leads precisely to a scenario in which the decision regarding the output sentence is guided by $p(\boldsymbol{\theta})$ (i.e., the prior distribution estimated on $\mathcal{T}$), while including a bias towards the adaptation data. Intuitively, the Bayesian framework has as benefit that the decision regarding the estimation of $\boldsymbol{\theta}$ is not taken by considering only the topic-specific data available (i.e., $\mathcal{A}$), which could lead to over-trained estimations. If the amount of such data is small, the parameter prior $p(\boldsymbol{\theta})$ will compensate this issue and provide robustness to the resulting estimation.

In this chapter, we will be focusing on adapting either the log-linear weights $\boldsymbol{\lambda}$ or the feature functions $\boldsymbol{h}$ present in state-of-the-art SMT systems (see Equation 1.6), since appropriate values of such parameters for a given domain do not necessarily imply a good combination in other domains. One naïve way in which some sort of adaptation can be performed is to re-estimate $\boldsymbol{\lambda}$ or $\boldsymbol{h}$ from scratch only on the adaptation data. However, this is usually not a good idea, since the amount of adaptation data available is usually not enough to provide stable estimations, yielding over-trained values of the model parameters that do not perform well in test time. In addition, such re-estimation is often not feasible given the high computational cost associated, which may range from a couple of hours to even days depending on the amount of adaptation data available. In this context, the Bayesian paradigm seems to be appropriate, since the inclusion of the prior over the model parameters should compensate the lack of data. In the work presented in the current chapter, *Bayesian predictive adapta-*

tion (BPA) will be used to solve both problems presented. Only $\boldsymbol{\lambda}$ or $\boldsymbol{h}$ will be adapted, alternatively. Nevertheless, the theory described here could also be used to adapt both.

In Section 4.2 a brief review of current approaches to translation model adaptation in SMT is provided. Although we are only aware of another work tackling adaptation in SMT from the Bayesian perspective, numerous works dealing with this problem by means of other methods have been reported in the literature. In addition, we also review some works applying BPA in other fields of natural language processing, such as speech recognition. Contributions of the present work are also explained in this section. Section 4.3 reviews briefly the Bayesian learning paradigm. Following these ideas, the formal derivation of BPA as applied to SMT is presented. We describe how to adapt both $\boldsymbol{\lambda}$ and $\boldsymbol{h}$, although in the present work we will not attempt to adapt both at the same time. In addition to analysing BPA in the most traditional case, as described above, we also study the possibility of extending the application to other similar scenarios: in Section 4.4, we analyse how to apply BPA in a scenario in which adaptation data is generated on the fly by a human expert who is amending the sentences produced by the system. Hence, it might well be the case that there is no adaptation data at all when the system is required to start translating, and adaptation has to take place in real time with the user interacting with the system. For such reason, adaptation time is critical, since it is not affordable for the human translator to be awaiting for the system to adapt and produce an adapted translation hypothesis. Another different scenario is considered in Section 4.5, namely, a scenario in which there is only a small amount of bilingual data available, both for $\mathcal{T}$ and $\mathcal{A}$. In such a scenario, state-of-the-art SMT systems become rather unstable, and small changes in the training or adaptation data have a very important impact on the final translation quality produced. Because of this, the main interest behind applying BPA in this case is to stabilise the parameters $\boldsymbol{\theta}$ estimated. Since Bayesian learning often implies computing the integral over the complete parametric space, sampling techniques are often used to solve this problem. The sampling methods studied in this chapter are presented in Section 4.6. Section 4.7 details the different practical approximations that need to be assumed before attempting to implement BPA within a real-world SMT system. Section 4.8 reports the experiments performed in order to assess how well the adaptation process performs in the different scenarios studied. For this purpose, n-best hypotheses provided by a state-of-the-art SMT system are re-ranked according to the Bayesian predictive distribution. Finally, conclusions of the present work and future research directions are detailed in Section 4.9.

4.2 Related work

In addition to the works described in Chapter 3 concerning language model adaptation in SMT and other approaches tackling adaptation from the mixture model point of view, there are other numerous works that confront the problem of translation model adaptation from different perspectives. For instance, Foster et al. (Foster et al., 2010) apply instance weighting techniques in order to weight out-of-domain phrase pairs according to the similarity of such phrase to the specific domain, and establishing whether it belongs to general language. The weights are established by means of a logistic model which takes into account simple features such as the number of tokens of that specific phrase pair, its frequency, the number of out-of-vocabulary words it contains, etc. In doing so, they show that it is possible to achieve

consistent improvements. Another strategy that has been applied for adapting SMT systems is to mine unseen words from dictionaries (Daume III and Jagarlamudi, 2011). In this sense, the authors improve translation results by extracting translations from other domains for those words that are considered out-of-vocabulary by the system for that specific domain. Finally, other works attempt to perform domain adaptation by selecting as training corpus only those sentences belonging to a large collection of data that seem to be important in the specific domain tackled (Axelrod et al., 2011; Gascó et al., 2010).

With respect to Bayesian methods applied to SMT, Zhang et al. (Zhang et al., 2008) apply Bayesian learning in order to estimate appropriate word-alignments within a synchronous grammar. Similarly, replacing the expectation-maximisation (EM) algorithm by Bayesian inference has also been studied (Mermer and Saraclar, 2011), and results have shown that applying Bayesian inference by means of a Gibbs sampler leads to interesting improvements in translation quality. Furthermore, the proposed method is also shown to overcome a common problem with EM-estimated word-alignment models, namely, that rare words tend to accumulate too much probability mass. The phrase-alignment problem has also been researched under the Bayesian learning paradigm (DeNero et al., 2008). In that work, the authors develop a phrase extraction algorithm that does not depend on a heuristic process, but rather attempts to extract the phrases through sampling from a translation model including Bayesian prior information by means of a Gibbs sampler. Recently, the Bayesian learning paradigm was applied with the purpose of adapting the word alignments that are included in most state-of-the-art SMT systems (Duh et al., 2011). In that work, the authors propose the use of sequential Bayesian methods with the purpose of adapting alignment models estimated on a broad domain corpus to a more specific domain, showing consistent improvements among different language pairs. In the present work, however, our purpose is to adapt the parameters of the final phrase-based model directly, and not the parameters of the single-word models that precede the estimation of the phrase-based model. Lastly, Bayesian inference has also been applied successfully to decipherment (Ravi and Knight, 2011), which is a problem closely related to SMT.

Although only recently applied to SMT, Bayesian adaptation has been broadly and successfully applied in other natural language processing areas, such as speech recognition (Huo et al., 1995; Kenny et al., 2000; Yu and Gales, 2005). In fact, work done in this direction is very broad, covering both batch adaptation (Yu and Gales, 2005) and online adaptation (Yu and Gales, 2006). Variational Bayes approaches have also been studied (Valente and Wellekens, 2005), which attempt to find a lower bound to approximate the intractable marginal likelihood (i.e., the likelihood where model parameters have been marginalised), yielding point estimates of the model parameters. An alternative to variational Bayes consists in approximating the marginal likelihood directly by sampling from the posterior distribution of the data given the model parameters (Yu and Gales, 2005), yielding an approximation of the real distribution, rather than a point estimate. This latter approach is often referred to as Bayesian predictive adaptation (BPA), and usually leads to more robust estimates. This is the approach that will be followed in the present work.

The present chapter extends work already published in adaptation in SMT in the following aspects:

- Bayesian predictive adaptation is presented as an appropriate formal framework for conducting SMT model adaptation.

- Positive results concerning the adaptation of either log-linear weights or feature functions are presented.
- Different sampling strategies are analysed for their application in Bayesian predictive adaptation.
- An online version of Bayesian predictive adaptation is shown to have an appropriate behaviour when adapting the log-linear weights.
- Finally, Bayesian predictive adaptation is also used in order to provide more robustness to the log-linear weights $\boldsymbol{\lambda}$ of a state-of-the-art SMT system trained in low-resource conditions.

Note that the work presented in this chapter is compatible with much of the work presented so far concerning adaptation in SMT. For instance, BPA may be applied in combination with the language model adaptation technique presented in Chapter 3 or different data combination strategies (see Section 3.2). However, in the present chapter the purpose is to analyse the performance of the Bayesian predictive adaptation strategies presented, leaving such combination experiments for future work.

4.3 Bayesian predictive adaptation for SMT

The process of adaptation can be viewed as a statistical process in which some prior knowledge exists regarding the estimation of the model parameters, but there is still some uncertainty about what the exact best estimation might be. In other words, a canonical model with parameters $\boldsymbol{\theta}_{\mathcal{T}}$ is already available, and it can be assumed that such estimation is a robust estimation obtained from a large collection of data. Then, as further evidence arrives, we would like that such estimations are revised so that they reflect the newly arrived data. Such is the case in the Bayesian learning paradigm (Bishop, 2006; Duda et al., 2001), where model parameters are viewed as random variables having some kind of a priori distribution. Observing these random variables leads to a posterior density, which sharpens with additional observations, and which typically peaks at the optimal values of the model parameters.

An important advantage of the Bayesian learning paradigm is that it allows to incorporate prior knowledge in the form of a parameter prior. By doing so, it is able to provide robust parameter estimates whenever the evidence provided by the training data (or adaptation data in this case) is not significant enough, i.e., the amount of training (adaptation) data is small. However, the effect of such prior knowledge fades when incorporating further evidence to our training data, until a point in which the contribution of the parameter prior towards the complete model distribution is negligible. In addition, the Bayesian learning paradigm does not attempt to obtain a single best point estimate of the model parameters, but rather relies on considering all possible parameter values, allowing uncertainty regarding what the best estimations of such parameters might be.

Within the Bayesian learning paradigm, the probability $p(\boldsymbol{y} \mid \boldsymbol{x})$ within Equation 1.1 can be reformulated by means of the predictive distribution as

$$p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}) \quad = \quad \int p(\boldsymbol{y}, \boldsymbol{\theta} \mid \boldsymbol{x}; \mathcal{T}) d\boldsymbol{\theta}, \tag{4.1}$$

where $\mathcal{T}$ represents the complete training set and $\boldsymbol{\theta}$ are the model parameters.

However, since we are interested in Bayesian *adaptation*, we need to consider one training set $\mathcal{T}$ and one adaptation set $\mathcal{A}$, leading to

$$
\begin{aligned}
p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}) &= \int p(\boldsymbol{y}, \boldsymbol{\theta} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}) d\boldsymbol{\theta} && (4.2)\\
&= \int p(\boldsymbol{\theta} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A})\; p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}; \mathcal{T}, \mathcal{A}) d\boldsymbol{\theta} && (4.3)\\
&\approx \int p(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{A})\; p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) d\boldsymbol{\theta}. && (4.4)
\end{aligned}
$$

From Equation 4.3 to Equation 4.4 it has been assumed, on the one hand, that the probability of the output sentence $\boldsymbol{y}$ does not depend on the complete training and adaptation data, whenever the model parameters $\boldsymbol{\theta}$ are known. On the other hand, it has also been assumed that the model parameters are independent from the actual input sentence $\boldsymbol{x}$. Such simplifications lead to a decomposition of the integral in two parts: the first one, $p(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{A})$ will assess how good the current model parameters are, and the second one, $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$, will account for the quality of the translation $\boldsymbol{y}$ given the current model parameters. In addition, the integral over the complete parametric space will force the model to take into account all possible values of the model parameters, although the prior over the parameters will bias the final distribution towards those values which are closer to our prior knowledge.

Operating with the probability of $\boldsymbol{\theta}$ by means of the Bayes' rule, we obtain:

$$
p(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{A}) = \frac{p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})\; p(\boldsymbol{\theta} \mid \mathcal{T})}{\int p(\mathcal{A} \mid \boldsymbol{\theta}'; \mathcal{T})\; p(\boldsymbol{\theta}' \mid \mathcal{T})\; d\boldsymbol{\theta}'}. \tag{4.5}
$$

In order to simplify Equation 4.5, and focusing on the probability of the adaptation data $\mathcal{A}$, of size $|\mathcal{A}|$, we obtain:

$$
\begin{aligned}
p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) \approx p(\mathcal{A} \mid \boldsymbol{\theta}) &= \prod_{a=1}^{|\mathcal{A}|} p(\boldsymbol{x}_a, \boldsymbol{y}_a \mid \boldsymbol{\theta}) && (4.6)\\
&= \prod_{a=1}^{|\mathcal{A}|} p(\boldsymbol{x}_a \mid \boldsymbol{\theta})\; p(\boldsymbol{y}_a \mid \boldsymbol{x}_a, \boldsymbol{\theta}), && (4.7)
\end{aligned}
$$

where the probability of the adaptation data has been assumed to be independent of the training data given that $\boldsymbol{\theta}$ is known and has been modelled as the probability of each bilingual sample $(\boldsymbol{x}_a, \boldsymbol{y}_a) \in \mathcal{A}$ being generated independently by a given translation model.

For modelling the prior over the model parameters, i.e., $p(\boldsymbol{\theta} \mid \mathcal{T})$, we will assume that the model parameters follow a normal distribution centred on $\boldsymbol{\theta}_{\mathcal{T}}$, i.e., the parameter values estimated on the training data, and with a diagonal covariance matrix $I \cdot \sigma_{\mathcal{T}}$, yielding

$$
p(\boldsymbol{\theta} \mid \mathcal{T}) \sim \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_{\mathcal{T}}, I \cdot \sigma_{\mathcal{T}}) = \frac{1}{(2\pi)^{d/2}\; |\sigma_{\mathcal{T}}|^{1/2}}\; \exp\left\{ -\frac{||\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathcal{T}}||^2}{2\sigma_{\mathcal{T}}} \right\}, \tag{4.8}
$$

with the variance $\sigma_{\mathcal{T}}$ assumed to be uniform for all parameters. Although there might be reasons for considering a full covariance matrix instead of a diagonal one, or even a co-variance

matrix where the diagonal is not constant, in the present thesis $\sigma_{\mathcal{T}}$ will be considered bounded for all parameters. Such co-variance matrix, which could be estimated by means of a development set, is left as a possible generalisation to the present work. d is the dimensionality of $\boldsymbol{\theta}$, i.e., the number of parameters that are going to be adapted. In this Equation, as in the rest of the present thesis, symbol $\sim$ means that $p(\boldsymbol{\theta} \mid \mathcal{T})$ is distributed following a certain distribution, which is specified to the right of such symbol. For now, and in order to preserve generality, we will not instantiate parameters $\boldsymbol{\theta}$.

To summarise, $p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A})$ is given by expression

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}) &\approx \mathcal{Z} \int p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})\, p(\boldsymbol{\theta} \mid \mathcal{T})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, d\boldsymbol{\theta} \\ &\approx \mathcal{Z} \int \prod_{a=1}^{|\mathcal{A}|} p(\boldsymbol{y}_a \mid \boldsymbol{x}_a, \boldsymbol{\theta})\, \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_{\mathcal{T}}, I \cdot \sigma_{\mathcal{T}})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, d\boldsymbol{\theta}. \quad (4.9) \end{aligned}$$

Here, $\mathcal{Z}$ is the normalisation constant required for ensuring that $p(\boldsymbol{y}|\boldsymbol{x}; \mathcal{T}, \mathcal{A})$ is actually a probability. The term $p(\boldsymbol{x}_a \mid \boldsymbol{\theta})$ present in Equation 4.7 can be simplified if $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})$ is plugged into Equation 4.5. The intuitive reason for this lies in the definition of the model of $p(\boldsymbol{y} \mid \boldsymbol{x})$ itself. One of the main advantages of using a conditional model $p(\boldsymbol{y} \mid \boldsymbol{x})$ (discriminative model), instead of attempting to model the joint distribution $p(\boldsymbol{y}, \boldsymbol{x})$ (generative model), is that the conditional model does not need to include a model of $p(\boldsymbol{x})$ (Sutton and McCallum, 2006). For this reason, $p(\boldsymbol{x})$ can be assumed to be independent of the model parameters $\boldsymbol{\theta}$. Hence, the term $p(\boldsymbol{x}_a \mid \boldsymbol{\theta})$ present in Equation 4.7 can be out-factored in the integral in the denominator of Equation 4.5, and then simplified with the same term in the numerator.

Note that the formulation presented here is general enough so as to consider as model parameters both the log-linear weights $\boldsymbol{\lambda}$ and the feature functions $\boldsymbol{h}(\cdot, \cdot)$ detailed in Equation 1.6. In the following, a detailed formulation about how to apply BPA to the log-linear weights $\boldsymbol{\lambda}$ or, alternatively, to the feature functions, is presented. Even though the formulation allows considering both as parameters, and the formulation required for adapting both is pretty straight-forward once the formulation for adapting each independently is available, we will not attempt to adapt both in the present work. The reason for this is that, as it will be analysed later on, adapting the feature functions is already a very sparse and computationally costly problem. Hence, adapting both together is a problem that still requires much more research before being able to yield satisfactory results.

If the adaptation data is known beforehand, i.e., there is a bilingual set of data that may be used for adaptation purposes before the actual test needs to be translated, the BPA procedure may take place in an offline setting, in which computational restrictions are not so demanding. We will name this kind of adaptation *batch* adaptation, and in such case the above formulae can be applied directly. Alternatively, if there is no adaptation set readily available before the actual test set is to be translated, it is also possible to use the test sentences that have already been translated as adaptation data for the next sentences to be translated, assuming an interactive scenario in which each sentence is corrected and validated by a human user immediately after such sentence is translated by the system. Thus, the adaptation data is viewed by the system as a data stream, in which each sample arrives at a given time t and the system needs to make the best out of the information it contains. This kind of scenario

will be called *online* adaptation, and in this case computational restrictions are much more important given that the system needs to adapt its parameters in real time. Section 4.4 will be devoted to instantiating the formulae presented in this section to an online scenario.

4.3.1 Adaptation of log-linear weights

One way to cope with the adaptation problem is to adapt the scaling weights $\boldsymbol{\lambda}$ present in state-of-the-art SMT systems, described in Equation 1.6 as

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m \, h_m(\boldsymbol{x}, \boldsymbol{y}) = \underset{\boldsymbol{y}}{\operatorname{argmax}} \, \boldsymbol{\lambda} \cdot \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}).$$

These weights adjust the importance of each single model within the specific task being dealt with. However, good values for a certain task might not be appropriate values for other tasks. To exemplify this, consider for instance that the original translation model has been trained on a domain in which sentences tend to be long, such as for example in a parliamentary debate. Then, if we intend to translate another domain in which sentences are rather short, such as sentences of medical diagnosis, we would ideally like that $\boldsymbol{\lambda}$ is adjusted conveniently to reflect this fact. Obviously, adapting $\boldsymbol{\lambda}$ will have the drawback that the individual models (i.e., the features $\boldsymbol{h}(\cdot,\cdot)$) will not be adapted to the new task, and, furthermore, unseen events that did not appear in $\mathcal{T}$ but do appear in $\mathcal{A}$ will still be considered unseen by the adapted model. Nevertheless, and although adapting $\boldsymbol{\lambda}$ is a coarse-grained adaptation strategy, it cannot be underestimated, since adjusting the importance of every single model present in state-of-the-art SMT systems often leads to very large improvements in the final translation quality delivered by the system. Adapting $\boldsymbol{\lambda}$ could be seen as an efficient adaptation strategy aimed at adaptation between tasks which are different, but not dramatically different. When attempting to adapt a translation model to a very different task, adapting $\boldsymbol{\lambda}$ might possibly not be enough, since e.g. out-of-vocabulary words will have a much more important effect than $\boldsymbol{\lambda}$.

Typically, the weights of the log-linear combination in Equation 1.6 are optimised by means of *minimum error rate training* (MERT) (Och, 2003), as described in Section 1.2.1. Such algorithm consists of two basic steps. First, n-best hypotheses are extracted for each one of the sentences of a given development set. Next, the optimum $\boldsymbol{\lambda}$ is computed so that the best hypotheses in the n-best list, according to a reference translation and a given metric, are the ones that the search algorithm would produce. These two steps are repeated until convergence.

This approach has two main problems. On the one hand, that it heavily relies on having a fair amount of data available as development set. On the other hand, that it *only* relies on the data in the development set. These two problems have as consequence that, if the development set made available to the system is not big enough, MERT will most likely become unstable and fail in obtaining an appropriate weight vector $\boldsymbol{\lambda}$ (Clark et al., 2011; Gascó et al., 2010).

However, it is quite common to have a great amount of data available in a given domain, but only a small amount from the specific domain we are interested in translating. Precisely this scenario is appropriate for BPA: under this paradigm, the weight vector $\boldsymbol{\lambda}$ is *biased* towards the optimal one according to the adaptation set, while avoiding over-training towards

such set by not forgetting the generality provided by the training set. Furthermore, recomputing $\boldsymbol{\lambda}$ from scratch by means of MERT may imply a computational overhead which may not be acceptable in certain environments, such as SMT systems configured for on-line translation or interactive machine translation, in which the final human user is waiting for the translations to be produced.

For adapting the log-linear weights $\boldsymbol{\lambda}$ by means of BPA, Equation 4.9 needs to be instantiated by considering as translation model a log-linear model. Then, we can assume that the only parameters of our model are the log-linear weights $\boldsymbol{\lambda}$, i.e., $\boldsymbol{\theta} \equiv \boldsymbol{\lambda}$, and that the feature functions $\boldsymbol{h}$ are fixed. By doing so, we obtain

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, A) &= \mathcal{Z} \int p(\mathcal{A} \mid \boldsymbol{\lambda}; \mathcal{T})\, p(\boldsymbol{\lambda} \mid \mathcal{T})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\lambda}) d\boldsymbol{\lambda} \\ &\propto \int \prod_{a=1}^{|\mathcal{A}|} \frac{\exp \sum_m \lambda_m h_m(\boldsymbol{x}_a, \boldsymbol{y}_a)}{\sum_{\boldsymbol{y}'} \exp \sum_m \lambda_m h_m(\boldsymbol{x}_a, \boldsymbol{y}')} \\ &\quad \exp\left\{ -\frac{||\boldsymbol{\lambda} - \boldsymbol{\lambda}_{\mathcal{T}}||^2}{2\sigma_{\mathcal{T}}} \right\} \\ &\quad \frac{\exp \sum_m \lambda_m h_m(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_m \lambda_m h_m(\boldsymbol{x}, \boldsymbol{y}')}\, d\boldsymbol{\lambda}, \end{aligned} \tag{4.10}$$

with the decision rule given by

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}). \tag{4.11}$$

It should be noted that the predictive distribution in Equation 4.10 includes, in its last term, the same distribution present in the original decision rule given in Equation 1.6, but complemented with the prior over the model parameters and the probability of the adaptation sample.

When taking a look at Equation 4.10, it is easy to think that the practical implementation of such formula will be too costly in computational terms. In fact, a common drawback when applying the Bayesian framework to real-sized tasks is precisely the computational expensiveness of the algorithms derived from such formulation. Nevertheless, we will see later on, in Section 4.6 that this issue can be efficiently handled by means of random sampling strategies, which will prove experimentally to yield an appropriate performance.

4.3.2 Adaptation of log-linear features

A natural extension of the adaptation of $\boldsymbol{\lambda}$ described above consists in adapting the log-linear feature functions $\boldsymbol{h} = \{h_1, \ldots, h_K\}$. However, adapting $\boldsymbol{h}$ is not an easy task, since such feature functions are often of a very different nature. For instance, some of them, as e.g. the translation models, are often defined at the local phrase level. This implies that the value of that specific locally-defined feature h for a given sentence pair $(\boldsymbol{x}, \boldsymbol{y})$ can be computed as the summation of the value of that feature for each one of the phrases $(\tilde{x}, \tilde{y})$ that compose that sentence pair, i.e., $h(\boldsymbol{x}, \boldsymbol{y}) = \sum_k h(\tilde{x}_k, \tilde{y}_k)$. Note that, in this case, we are using a summation instead of a product because the features are defined in the logarithmic domain.

However, other features, such as the reordering model, attempt to model long-range dependencies among phrases, and hence cannot be defined at the local phrase level. Other common features are the number of phrases present in the sentence and the number of words that compose the output sentence, which cannot be adapted. Although the theoretical framework would possibly be suitable for adapting all feature functions (which allow adaptation), and there could be reasons for doing so, in the present work we will only attempt to adapt those feature functions which can be defined at the local phrase level. Given the premise that such feature functions are defined at the phrase level, they can be considered either as functions assigning scores to certain phrase pairs, which is zero if such phrase pair has not been observed in training time, or as vectors containing the scores of the phrases seen in training time. Hence, the amount of parameters to be adapted in this case is usually in the range of several millions of parameters, i.e., the number of phrases that have been observed in training time multiplied by the number of features to be adapted. Let ℓ be the set of feature functions defined at the local phrase level. Then, instead of adapting each one of the feature functions in ℓ, we will simplify the problem by defining g as the weighted combination of such features, and attempt to adapt g instead. Formally, g is defined as

$$g(\boldsymbol{x},\boldsymbol{y}) = \sum_{l\in\ell} \lambda_l h_l(\boldsymbol{x},\boldsymbol{y}) = \sum_{l\in\ell}\sum_{k} \lambda_l h_l(\tilde{x}_k,\tilde{y}_k) = \sum_{k} g(\tilde{x}_k,\tilde{y}_k). \tag{4.12}$$

Then, in order to reduce sparseness, we can study the effect of adapting the translation models defined at the local level by adapting g. Hence, as done in Section 4.3.1, Equation 4.9 can be instantiated considering $\boldsymbol{\theta} \equiv \mathbf{g}$ as

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{x};\mathcal{T},\mathcal{A}) &= \mathcal{Z}\int p(\mathcal{A} \mid \mathbf{g};\mathcal{T})p(\mathbf{g} \mid \mathcal{T})p(\boldsymbol{y} \mid \boldsymbol{x},\mathbf{g})d\mathbf{g} \\ &\propto \int \prod_{a=1}^{|\mathcal{A}|} \frac{\exp\left\{g(\boldsymbol{x}_a,\boldsymbol{y}_a) + \sum_{m\notin\ell}\lambda_m h_m(\boldsymbol{x}_a,\boldsymbol{y}_a)\right\}}{\sum_{\boldsymbol{y}'}\exp\left\{g(\boldsymbol{x}_a,\boldsymbol{y}') + \sum_{m\notin\ell}\lambda_m h_m(\boldsymbol{x}_a,\boldsymbol{y}')\right\}} \\ &\quad \exp\left\{-\frac{||\mathbf{g}-\mathbf{g}_{\mathcal{T}}||^2}{2\sigma_{\mathcal{T}}}\right\} \\ &\quad \frac{\exp\left\{g(\boldsymbol{x},\boldsymbol{y}) + \sum_{m\notin\ell}\lambda_m\, h_m(\boldsymbol{x},\boldsymbol{y})\right\}}{\sum_{\boldsymbol{y}'}\exp\left\{g(\boldsymbol{x},\boldsymbol{y}') + \sum_{m\notin\ell}\lambda_m\, h_m(\boldsymbol{x},\boldsymbol{y}')\right\}}\, d\mathbf{g}. \end{aligned} \tag{4.13}$$

In this last equation, the term $||\mathbf{g}-\mathbf{g}_{\mathcal{T}}||^2$ present in the parameter prior is well defined if $\mathbf{g}$ is understood as a vector with the size of the phrase-table.

As explained in Section 4.3.1, in this case the integral in Equation 4.13 will also be handled by means of random sampling strategies. However, it must also be considered that in the case of Section 4.3.1 the size of $\boldsymbol{\theta}$ (in that case, the log-linear weights $\boldsymbol{\lambda}$) is very small, since the amount of models included in state-of-the-art log-linear models is usually around 14. In contrast, when adapting the feature functions the size of $\boldsymbol{\theta}$ (in this case the features $\mathbf{g}$ defined at the local translation unit level) is much larger, in the range of several millions of parameters. For this reason, and as will be explained more in detail in Section 4.8, the adaptation of the feature functions under the BPA paradigm will not be as successful as the adaptation of the log-linear weights $\boldsymbol{\lambda}$.

4.4 Online Bayesian adaptation

The adaptation problem becomes even more important in scenarios where collaboration between a human expert and a machine translation system is required in order to achieve high quality translations in an efficient manner. This is the case in scenarios such as *computer assisted translation* (CAT) and *interactive machine translation* (IMT) (Barrachina et al., 2009) (see Section 1.3), where human-machine interaction is essential to produce high quality results while profiting of the efficiency of machine translation systems. In these scenarios, the SMT system proposes a translation hypothesis to the human expert, who may then amend the sentence or accept it completely as correct. Then, the human translator expects the system to learn from its own errors and improve its future translations by using the feedback provided. To make this problem even more challenging, it is often the case that human translators need to translate documents with different styles and topics, even in the same day. For this reason, two main challenges arise: first, to make use of the adaptation data provided by the user even when such adaptation data is very scarce because he has just started working on a new domain. Second, to perform adaptation based on the current input data, which might be different from the data collected previously, implying that parameters computed for the first set of sentences might not be appropriate for subsequent ones. These two problems are specially adequate for an *online* implementation of BPA, given that the stability will be provided by the prior over the model parameters. However, since adapting the feature functions is way too costly for an online setting, in this case we will only attempt to adapt the log-linear weights $\boldsymbol{\lambda}$. By considering as adaptation set $\mathcal{A} = \mathcal{A}_t$ only the last $|\mathcal{A}_t|$ sentences already corrected by the human translator at time t, and considering as model parameters $\boldsymbol{\lambda} \equiv \boldsymbol{\theta}$, the BPA paradigm may also be applied for online adaptation by instantiating Equation 4.9 as follows:

$$
\begin{aligned}
p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}_t) &= \mathcal{Z} \int p(\mathcal{A}_t \mid \boldsymbol{\lambda}; \mathcal{T}) p(\boldsymbol{\lambda} \mid \mathcal{T}) p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\lambda}) d\boldsymbol{\lambda} \\
&\propto \mathcal{Z} \int \prod_{a=1}^{|\mathcal{A}_t|} \frac{\exp \sum_k \lambda_k \, h_k(\boldsymbol{x}_a, \boldsymbol{y}_a)}{\sum_{\boldsymbol{y}'} \exp \sum_k \lambda_k \, h_k(\boldsymbol{x}_a, \boldsymbol{y}')} \\
&\quad \exp\left\{ -\frac{||\boldsymbol{\lambda} - \boldsymbol{\lambda}_{\mathcal{T}}||^2}{2\sigma_{\mathcal{T}}} \right\} \\
&\quad \frac{\exp \sum_k \lambda_k \, h_k(\boldsymbol{x}, \boldsymbol{y})}{\sum_{\boldsymbol{y}'} \exp \sum_k \lambda_k \, h_k(\boldsymbol{x}, \boldsymbol{y}')} \, d\boldsymbol{\lambda}.
\end{aligned} \tag{4.14}
$$

Note that the data within $\mathcal{A}_t$ may be as small as one sentence, or even only an incomplete sentence. In the case $|\mathcal{A}_t| = 1$, we have that the system may already start with the online adaptation with as few as one adaptation sentence. Furthermore, if the SMT system is being used within an interactive environment, such as IMT, $|\mathcal{A}_t|$ may even be less than one: whenever a human translator has validated part of the sentence that is being translated, the SMT system may already start the adaptation process by using as new evidence the chunk of sentence that has already been validated. $\mathcal{A}_t$ could be seen as a sort of cache, or trailing sliding window, whose purpose is to bias the model distribution towards the data seen more recently.

As will be seen later in Section 4.6, in the Bayesian framework it is quite typical to replace the integral over the complete parametric space by a random sampling. Assuming such

sampling fixed, Equation 4.14 allows an efficient, incremental implementation. To understand why this is so, let us analyse each component independently: first, $p(\boldsymbol{\lambda} \mid \mathcal{T})$ can be precomputed. Second, $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\lambda})$ needs to be computed for each test sentence, and for each hypothesis considered, including the summation in the denominator. However, once $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\lambda})$ has been computed, $p(\mathcal{A}_t \mid \boldsymbol{\lambda}; \mathcal{T})$ only requires one division and one multiplication in order to incorporate the last sentence. Since each of the adaptation samples within $p(\mathcal{A}_t \mid \boldsymbol{\lambda}; \mathcal{T})$ were, at a given time, test sentences, incorporating the probability of the sentence seen at time $t-1$ into $\mathcal{A}_t$ only requires one multiplication and one division. Hence, applying Eq. 4.9 in an on-line setting does not require a significant computational overhead when compared to the cost of performing the search for the output sentence.

4.5 Bayesian adaptation for model stabilisation

Although the main goal of BPA is model adaptation, another possible application of BPA is for model stabilisation, where the main goal is to achieve a model that is less prone to overtrain towards specific characteristics of the training set provided. This is quite frequent when training data is scarce. In the model adaptation task, it is assumed that there is a large amount of bilingual data readily available from a given domain, but only few data from the specific domain we are interested in translating. However, it is not always reasonable to assume that such large amount of data is available in order to obtain a good estimation for $\boldsymbol{\theta}_{\mathcal{T}}$. In some tasks, such as the recent Haitian Creole translation task [a], the amount of data available for that specific translation pair is very scarce, and techniques must be developed for avoiding model over-training, which would lead to an unstable system in translation time. BPA can also be applied under this framework, with the purpose of alleviating the problems derived from data scarcity. In this work, we will be exploring the stabilisation of the log-linear model weights $\boldsymbol{\lambda}$, whose estimation has been shown to be critical and reportedly unstable (Clark et al., 2011; Gascó et al., 2010). Specifically, we explored two different possibilities:

- Assume that a small development set $\mathcal{D} \subset \mathcal{T}$ is available. This development set may not be enough to obtain a good estimation of $\boldsymbol{\lambda}_{\mathcal{T}}$, but may be enough to be used as mean vector for the Gaussian parameter prior within BPA. Then, the sampling procedure will account for taking into consideration the neighbouring points within the parameter hyperplane, thus allowing the SMT system to consider a wider range of different parameters. Hence, the parameter prior is given by expression $p(\boldsymbol{\lambda} \mid \mathcal{T}) \sim \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\lambda}_{\mathcal{D}}, I \cdot \sigma_{\mathcal{D}})$, with $\boldsymbol{\lambda}_{\mathcal{D}}$ and $\sigma_{\mathcal{D}}$ being estimated on set $\mathcal{D}$.

- Assume there is no appropriate development set at all, or that the set that would be used as development set would be best used as training data, or even that such development set is available and it is possible to obtain a certain $\boldsymbol{\lambda}_{\mathcal{D}}$, but this $\boldsymbol{\lambda}_{\mathcal{D}}$ is not an appropriate value for the mean of the Gaussian prior. However, we will assume that there is some canonical set of parameters $\boldsymbol{\lambda}_C$, which was obtained beforehand in some way which is not important at this point (i.e., a very different task), but which is considered to be robust enough. In this case, $p(\boldsymbol{\lambda} \mid \mathcal{T}) \sim \mathcal{N}(\boldsymbol{\lambda}; \boldsymbol{\lambda}_C, I \cdot \sigma_C)$.

[a] www.statmt.org/wmt11/featured-translation-task.html

Input: $\boldsymbol{\theta}_{\mathcal{T}}$, the parameter mean vector of size Q

Output: $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$, a (pseudo-)random sampling of $\boldsymbol{\theta}_{\mathcal{T}}$

Initialise: $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}}) = \{\boldsymbol{\theta}_{\mathcal{T}}\}$

For i **in** $\{1, \ldots, N_s\}$ **do:**

$\mathbf{s} = \boldsymbol{\theta}_{\mathcal{T}}$

$k = i \bmod Q$

$s_k = s_k + \mathrm{rand}(-0.5, 0.5)$

$\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}}) = \mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}}) \cup \{\mathbf{s}\}$

Figure 4.1: Algorithm for performing the heuristic sampling described. $\mathrm{rand}(a, b)$ is a value drawn randomly in the interval $[a, b]$, N_s is the desired size of $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ and $\mathbf{s} = [s_1, \ldots, s_k]^T$ is a single sample.

4.6 Sampling methods

Although Equation 4.9 is the correct thing to do from a theoretical point of view, in practise computing the integral over the complete parametric space is unfeasible from the computational point of view. Moreover, it may also be the case that the function to be integrated is not even integrable. For this reason, it is quite common to approximate such integral by means of a discrete sum over a sampling of such parameters. In this chapter, several sampling techniques are explored, ranging from a simple heuristic to the statistically sound Metropolis-Hastings algorithm (Hastings, 1970).

In order to preserve generality, the sampling methods described in the following will be formulated in terms of $\boldsymbol{\theta}$, which may be appropriately instantiated according to Sections 4.3, 4.4, and 4.5. In the following, a specific sampling of $\boldsymbol{\theta}$ will be denoted by $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$. Although some of the algorithms presented for obtaining $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ will actually depend on other variables aside $\boldsymbol{\theta}_{\mathcal{T}}$, $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ is adopted for denoting a generic sample of $\boldsymbol{\theta}$, and $\mathcal{S}_{p(\cdot)}(\boldsymbol{\theta}_{\mathcal{T}})$ is employed for denoting that the sample has been obtained according to distribution $p(\cdot)$. This subindex will be dropped especially in the experiments section, Section 4.8, with the purpose of keeping notation unclogged and whenever such subindex can be assumed.

4.6.1 Heuristic sampling

As a first approach to sampling the integral in Equation 4.9, the close neighbourhood of the mean vector of the parameter prior was explored. For doing this, each one of the components of the parameter vector was perturbed by a random amount, successively, as described in Figure 4.1. In this case, $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ does not include any subindex because the distribution from which it has been obtained is unknown and relies on pure heuristic decisions motivated by working well in practise.

Once an appropriate sample $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ has been obtained, Equation 4.9 is approximated in

this case as

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}) &= \mathcal{Z} \int p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) p(\boldsymbol{\theta} \mid \mathcal{T}) p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) \, d\boldsymbol{\theta} \\ &\approx \mathcal{Z}' \sum_{\boldsymbol{\theta} \in \mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})} p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) p(\boldsymbol{\theta} \mid \mathcal{T}) p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}). \end{aligned} \tag{4.15}$$

Note that normalisation constant $\mathcal{Z}$ has been replaced by $\mathcal{Z}'$ so that $p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A})$ is still an appropriately normalised probability.

Although this algorithm obviously involves a series of heuristic decisions and does not depend on the actual probability to be sampled, it has one main advantage: it is independent from $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})$. This means that most of the terms within the integral in Equation 4.15 can be precomputed, except for the probability of the current test sentence, i.e., $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$. Obviously, this implies that $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ does not need to be recomputed whenever a new adaptation sample arrives, which would be far too costly when applying BPA in an online scenario.

However, this heuristic algorithm has an important drawback. Other sampling algorithms, such as Markov chain Monte-Carlo (Bishop, 2006), are appropriate for sampling from unnormalised distributions, but the algorithm presented here is sensible to normalisation. This can be seen e.g. in Equation 4.10: dropping normalisation constants leads to a product of probabilities when computing the probability of the adaptation sample, which implies that larger amounts of adaptation data will lead to smaller numeric values. Hence, increasing the size of $\mathcal{A}$ might fail to bias the final integral in a more stronger fashion when compared to the prior $p(\boldsymbol{\theta} \mid \mathcal{T})$. For this reason, Equation 4.15 is complemented with a leveraging factor δ, such that

$$p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}) = \sum_{\boldsymbol{\theta} \in \mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})} \left(p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\right)^{\frac{1}{\delta}} p(\boldsymbol{\theta} \mid \mathcal{T}). \tag{4.16}$$

Although there are other ways of adding this leveraging term, we chose this one for numeric reasons.

Note that, although this algorithm resembles slightly the Gibbs sampling procedure (Geman and Geman, 1984), there are important differences. In Gibbs sampling, each one of the components of $\boldsymbol{\theta}$ would be drawn from the distribution $p(\theta_k \mid \boldsymbol{\theta}_{\setminus k})$, where $\boldsymbol{\theta}_{\setminus k}$ denotes $\theta_1, \ldots, \theta_K$ but with θ_k omitted. This drawing procedure is repeated by cycling through each one of the components, but when a new sample is drawn, the new value of θ_k is used for drawing the next sample, hence building a Markov chain. This is not the case in the algorithm presented above. Hence, it cannot be said to form a Markov chain and it is not guaranteed that it will finally sample from the desired distribution. In addition, from a pure theoretical point of view, the term $p(\boldsymbol{\theta} \mid \mathcal{T})$ should be removed from Equation 4.15 (and hence also from Equation 4.16) whenever it can be assumed that the heuristic sampling method described here is obtaining a sample of $p(\boldsymbol{\theta} \mid \mathcal{T})$. However, experimental results show that it presents an appropriate behaviour for the specific task tackled here.

4.6.2 Gaussian sampling

The algorithm described in the previous section has the advantage that it does not require the adaptation set to be known beforehand, and hence leads to the benefit of being able to

precompute most of the terms in Equation 4.15. On the other hand, it is a heuristic approach that requires the introduction of an additional parameter. An alternative approach that is still independent from the adaptation data, but has a closer relation to the actual probability being sampled is to sample the normal distribution (i.e., the parameter prior) directly, without taking into account the probability of the adaptation data. In this way, the samples obtained will follow the distribution $p(\boldsymbol{\theta} \mid \mathcal{T}) \sim \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_{\mathcal{T}}, I \cdot \sigma_{\mathcal{T}})$, which implies that they are more closely related to the actual probability that should be sampled. However, it is not necessary to re-compute the parameter sampling whenever new adaptation data arrives, as is the case with the sampling strategy to be presented in the next section.

When sampling $\boldsymbol{\theta}$ according to $p(\boldsymbol{\theta} \mid \mathcal{T})$, then Equation 4.9 can be approximated, by the Strong Law of Large Numbers (Robert and Casella, 2004), as

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}) &= \mathcal{Z} \int p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) p(\boldsymbol{\theta} \mid \mathcal{T}) p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) \, d\boldsymbol{\theta} \\ &\approx \mathcal{Z}' \sum_{\boldsymbol{\theta} \in \mathcal{S}_{p(\boldsymbol{\theta} \mid \mathcal{T})}(\boldsymbol{\theta}_{\mathcal{T}})} p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}), \end{aligned} \tag{4.17}$$

where the approximation will be an equality for $|\mathcal{S}_{p(\boldsymbol{\theta}|\mathcal{T})}(\boldsymbol{\theta}_{\mathcal{T}})| \to \infty$.

Looking at Equation 4.17 makes it obvious that considering a δ leveraging factor is meaningless when considering Gaussian sampling, since the prior $p(\boldsymbol{\theta} \mid \mathcal{T})$ is not even present in the summation.

4.6.3 Markov chain Monte Carlo

The purpose of *Markov chain Monte Carlo* (MCMC) methods (Bishop, 2006) is to obtain a set of samples $\mathcal{S}_{p(\cdot)}(\boldsymbol{\theta}_{\mathcal{T}})$ of a variable (in this case $\boldsymbol{\theta}$), where each sample is assumed to be drawn from a certain distribution $p(\cdot)$, in this case the one comprised within the integral in Equation 4.4, i.e., $p(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{A})$. MCMC methods are widely used in the machine learning community when applying Bayesian methods and are specially appropriate for sampling from distributions where it is possible to evaluate such distribution except for a certain normalisation constant (Bishop, 2006). For doing this, a (first order) Markov chain is established, where each new sample[b] $\boldsymbol{\theta}^*$ depends on the previous sample $\boldsymbol{\theta}'$. Specifically, in this chapter we will be using the Metropolis-Hastings (MH) algorithm (Hastings, 1970).

The MH algorithm basically consists of two steps. First, a sample $\boldsymbol{\theta}^*$ from a given proposal distribution $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$ is drawn. Next, such sample is accepted with probability $A(\boldsymbol{\theta}^*, \boldsymbol{\theta}')$, given by expression

$$A(\boldsymbol{\theta}^*, \boldsymbol{\theta}') = \min\left(1, \frac{\tilde{p}(\boldsymbol{\theta}^*) q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}^*)}{\tilde{p}(\boldsymbol{\theta}') q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}')}\right), \tag{4.18}$$

with $p(\boldsymbol{\theta}) = \tilde{p}(\boldsymbol{\theta}) / \mathcal{Z}_p$ being the distribution from which we intend to sample ($p(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{A}) = p(\boldsymbol{\theta} \mid \mathcal{T}) p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) / \mathcal{Z}_p$ in this case), and $\mathcal{Z}_p$ being the normalisation term for $p(\boldsymbol{\theta})$. In Equation 4.18, it does not matter whether $\tilde{p}(\boldsymbol{\theta})$ is used instead of $p(\boldsymbol{\theta})$, since the normalisation

[b] Typically, MCMC establishes a Markov chain between states of the Markov blanket, and the samples denoted here by $\boldsymbol{\theta}$ are actually states $\mathbf{z}$ of the Markov blanket. However, to simplify notation, in this chapter we assume $\mathbf{z} \equiv \boldsymbol{\theta}$.

term $\mathcal{Z}_p$ within $p(\boldsymbol{\theta})$ can be simplified and the resulting Markov chain would be identical. This is in practise very useful, since there are many applications, such as BPA in SMT, where $\mathcal{Z}_p$ cannot be computed. In addition, if the proposal distribution is symmetric, terms $q(\cdot \mid \cdot)$ can also be simplified.

Once an appropriate sample $\mathcal{S}_{p(\boldsymbol{\theta}|\mathcal{T},\mathcal{A})}(\boldsymbol{\theta}_\mathcal{T})$ of $p(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{A})$ has been obtained, Equation 4.9 is approximated, again by the Strong Law of Large Numbers (Robert and Casella, 2004), in this case as

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}) &= \mathcal{Z} \int p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) p(\boldsymbol{\theta} \mid \mathcal{T}) p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) \, d\boldsymbol{\theta} \\ &\approx \mathcal{Z}' \sum_{\boldsymbol{\theta} \in \mathcal{S}_{p(\boldsymbol{\theta}|\mathcal{T},\mathcal{A})}(\boldsymbol{\theta}_\mathcal{T})} p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}), \end{aligned} \tag{4.19}$$

where the approximation will be an equality for $|\mathcal{S}_{p(\boldsymbol{\theta}|\mathcal{T},\mathcal{A})}(\boldsymbol{\theta}_\mathcal{T})| \to \infty$. As in the case of Gaussian sampling, including a δ leveraging term when dealing with MCMC sampling is pointless.

Even though it might seem odd that term $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})$ is dropped in Equation 4.19, but not in Equation 4.17, the reason for this is that in the case of MCMC the $\boldsymbol{\theta}$-samples are obtained from the conjugate $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) \cdot p(\boldsymbol{\theta} \mid \mathcal{T})$, which is the same as obtaining them from the posterior density $p(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{A})$, since the normalisation term can be neglected safely because it is simplified in Equation 4.18. However, in the case of Gaussian sampling the $\boldsymbol{\theta}$-samples are extracted from $p(\boldsymbol{\theta} \mid \mathcal{T})$ directly, without taking into consideration the adaptation sample. In fact, dropping $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})$ in Equation 4.17 leads empirically to very bad results. In this context, it is also interesting to point out that, for $|\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})| \to \infty$, both methods should theoretically converge to the same distribution. Nevertheless, the different meta-parameters that control both sampling strategies may imply that one sampling strategy converges slower, as would be the case, e.g., if the MCMC chain gets stuck in a local optimum of the probability density function.

When building a MCMC chain, there are several things that need to be taken into account. In the first place, the proposal distribution q needs to be established. Quite often, this is done by setting

$$q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') \sim \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}', I \cdot \sigma_o), \tag{4.20}$$

where $\mathcal{N}(\boldsymbol{\theta}', I \cdot \sigma_o)$ is the normal distribution with mean vector $\boldsymbol{\theta}'$ and covariance matrix a diagonal matrix with main diagonal σ_o, whenever independence between the components of $\boldsymbol{\theta}$ can be assumed. However, establishing an appropriate σ_o is critical; on the one hand, because too small values of σ_o will lead to a high rejection rate and a slow mixing chain, meaning that the sampling chain will most likely get stuck at a local maximum of the density hyper-surface. On the other hand, because if σ_o is chosen too big it will lead to a chaotic chain which will keep moving back and forth and will not be able to sample the density function appropriately.

Another aspect that needs to be taken into account when building a MCMC chain is the burn-in phase, which is the number of samples that need to be drawn in order to be able to assume independence from the initial state of the Markov chain. This point may be very important, since if the starting point is not well chosen, the first samples obtained by the MCMC procedure may introduce a non-desired bias which does not depend on the distribution being sampled, but rather on the starting point of the Markov chain.

4.6.4 Viterbi-like approach

One last approach to the sampling problem is the Viterbi-like approach. Under this framework, the core idea is to approximate the integral in Equation 4.9 by

$$p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A}) = \prod_{a=1}^{|\mathcal{A}|} p(\boldsymbol{y}_a \mid \boldsymbol{x}_a, \hat{\boldsymbol{\theta}})\, \mathcal{N}(\hat{\boldsymbol{\theta}}; \boldsymbol{\theta}_{\mathcal{T}}, \sigma_{\mathcal{T}})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \hat{\boldsymbol{\theta}}) \tag{4.21}$$

where

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})}{\operatorname{argmax}} \prod_{a=1}^{|\mathcal{A}|} p(\boldsymbol{y}_a \mid \boldsymbol{x}_a, \boldsymbol{\theta})\, \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_{\mathcal{T}}, \sigma_{\mathcal{T}})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) \tag{4.22}$$

One important note regarding this kind of sampling is that, when assuming the Viterbi-like approach for the integral, the resulting formulation fits no longer into the Bayesian paradigm. The key aspect of the Bayesian framework is precisely that it does not rely on a single point-estimate of the model parameters, but rather keeps the generality provided by considering all possible parameters. When assuming a Viterbi-like approach, we are in fact assigning a single-best point estimate of the model parameters. Nevertheless, the Viterbi-like approach is, from an intuitive point of view, a very straight-forward approximation to the integral described in the BPA formulation, and, for this reason, we will also conduct experiments with this approach. It is worth noting that this single-best point estimate is still conceptually different from the single-best point estimate that would be obtained by applying the maximum-likelihood framework, or even by using MERT in the case of adapting $\boldsymbol{\lambda}$ (see Section 4.3), since in this Viterbi-like approach the parameter prior $p(\boldsymbol{\theta} \mid \mathcal{T})$ is still present, and this is not the case in non-Bayesian approaches.

The intuition behind the Viterbi-like approximation is that $p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A})$ could be, in fact, a very sharp distribution, having $\hat{\boldsymbol{\theta}}$ accumulate most of its probability mass. This is often the case in many natural language processing tasks, as for example in speech recognition. In other terms, this sampling approach could be seen as a sampling in which $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 1$, but with the specific $\mathbf{s} \equiv \mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ being chosen probabilistically according to distribution $p(\boldsymbol{y}, \boldsymbol{\theta} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A})$.

In this chapter, $\hat{\boldsymbol{\theta}}$ will be computed as the best $\boldsymbol{\theta}$ observed when sampling $p(\boldsymbol{y} \mid \boldsymbol{x}; \mathcal{T}, \mathcal{A})$ according to the algorithm described in Figure 4.1.

4.7 Practical approximations

In addition to performing a random sampling instead of computing the complete integral, there are several issues that need to be taken care of before attempting to implement the formula described in Equation 4.9 directly.

Firstly, the denominator within the components $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})$ and $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$ contains a sum over all possible sentences of the target language, which is not computable. For this reason, $\sum_{\boldsymbol{y}'}$ is approximated as the sum over all the hypothesis within the n-best list generated by the decoder. Moreover, instead of performing a full search of the best possible translation of a given input sentence, we will perform a re-rank of the n-best list provided by the decoder according to Equation 4.9.

In addition, typical state-of-the-art phrase-based SMT systems do not guarantee complete coverage of all possible sentence pairs due to the great number of heuristic decisions involved in the estimation of the translation models (see Section 1.2.1). Furthermore, out-of-vocabulary words may imply that the SMT model is unable to explain a certain bilingual sentence completely. This implies that the translation model is often unable to account for a source sentence having a fixed translation, as is the case in the adaptation data. Hence, computing $h(\boldsymbol{x}_a, \boldsymbol{y}_a)$ may not always be possible. For this reason, instead of using the true reference present in the adaptation set, we will be using the best possible translation that the system is able to provide, hence approximating $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})$ as

$$p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T}) \approx \prod_{a=1}^{|\mathcal{A}|} \frac{\exp \sum_k \lambda_k \; f_k(\boldsymbol{x}_a, \boldsymbol{y}_a^*)}{\sum_{\boldsymbol{y}'} \exp \sum_k \lambda_k f_k(\boldsymbol{x}_a, \boldsymbol{y}')}, \tag{4.23}$$

where $\boldsymbol{y}^*$ represents the best hypothesis the search algorithm is able to produce, according to a given translation quality measure. This approximation was assumed both when considering $\boldsymbol{\theta} \equiv \boldsymbol{\lambda}$ (Section 4.3.1) and $\boldsymbol{\theta} \equiv \mathbf{g}$ (Section 4.3.2), so that Equation 4.23 may be instantiated appropriately following to Equations 4.10, 4.13, and 4.14.

Note that, after the approximations described above, applying BPA for feature function adaptation as described in Section 4.3.2 implies that only those phrases already present in the phrase-table, i.e., phrases that have already been seen in the training data, may be affected by the BPA procedure. In order to introduce new phrases, it would be first necessary to solve the coverage problem described. This being done, it would be possible to introduce new phrases into the phrase-table with a certain ϵ score, and then allow the BPA procedure to determine whether that new phrase pair should be promoted. Theoretically, the formulation presented in Section 4.3 would allow the introduction of unseen phrases into the phrase-table with a (possibly small) score ϵ, and then allow the adaptation procedure to determine whether such phrase should gain more weight in the translation process. However, in the experiments performed in this chapter this was not done for comparison reasons, since our purpose is to analyse how well the BPA is able to adapt existing model parameters: introducing new phrases has already been done in other works (Ortiz-Martínez et al., 2010), and is known to provide interesting improvements.

Lastly, adapting $\boldsymbol{h}$ is a very costly operation, since the amount of parameters to be adapted is usually in the range of several millions. For this reason, instead of obtaining fully randomised parameter samples (i.e. sampling the whole g), we restrained such sampling to only those entries of $g(\tilde{x}, \tilde{y})$ that may actually produce a change in the translation of the test sentence being considered. This implies considering for adaptation only those phrase pairs that are present only in some of the translation hypotheses within the n-best list, but not in all of them. However, this is also costly, since it implies that, first, it must be assessed which phrases are to be considered. Then, parameter sampling needs to be performed once for each one of the sentences present in the test set. Note that, if the sampling of g is performed without constraints, it is most likely that $p(\boldsymbol{y} \mid \boldsymbol{x})$ no longer describes a probability distribution, since a re-normalisation step would be required. However, since the normalisation constant required would have no effect on the maximisation described in Equation 1.6, this re-normalisation step may be safely omitted.

4.8 Experiments

Experiments were performed by means of the open-source MT toolkit Moses (Koehn et al., 2007) in its default non-monotonic configuration. The phrase-tables were generated by means of symmetrised word alignments obtained with GIZA++ (Och and Ney, 2003). The language model used was a 5-gram with modified Kneser-Ney smoothing (Kneser and Ney, 1995), built with the SRILM toolkit (Stolcke, 2002). The log-linear combination weights in Equation 1.6 were optimised using *minimum error rate training* (MERT) (Och, 2003).

In this section, whenever a figure shows two plots side by side, the left plot will display translation quality and the right plot will display the corresponding confidence interval sizes. In addition, and unless stated otherwise, the x-axis will always be in logarithmic scale. The scale of the y-axis will be linear whenever the plot displays translation quality, and logarithmic in the case of the confidence interval sizes.

4.8.1 Corpora

The experiments conducted in this chapter were carried out on three different bilingual corpora, belonging each one to a different domain.

In the first place, the Europarl and the News-Commentary corpora, in their WMT10 partition, were considered (see Section 1.4 for further details on these corpora). Due to its generic nature, the Europarl corpus is suitable for training a first canonical SMT system, which will be then adapted to more specific tasks. Specifically, the standard features $\boldsymbol{h}$ were estimated on the training partition, whereas the log-linear combination weights $\boldsymbol{\lambda}$ were estimated on the development subset $\mathcal{D}$ by means of MERT. This set of weights will be referred to as $\boldsymbol{\lambda}_{\xi}$. In addition, the training part of the News-Commentary corpus will be used for the purpose of obtaining adaptation samples, which will be then used either as adaptation sample $\mathcal{A}$ within BPA, or as development set when re-estimating $\boldsymbol{\lambda}$ by means of MERT. Translation quality will be assessed on the NC 2009 test set.

Lastly, validation experiments were also conducted on the TED corpus. This corpus is obtained from a collection of public speeches on a variety of topics for which video, transcripts and translations are freely available on the Web. Again, the domain is very broad, since there is no restriction on the subject of the talks. However, due to the nature of the corpus, language style is very different from the other corpora mentioned. This corpus was used in a recent evaluation campaign (Paul et al., 2010), and is only available for French–English translation. Statistics are shown in Table 4.2. As for NC, the training part will be used for obtaining adaptation samples.

The major part of the experiments reported in this chapter were performed by using the NC 2009 set as test data, and hence with the adaptation data drawn at random from the NC training data. However, some experiments were also performed on the TED data, with the purpose of validating the conclusions drawn from the NC data. Hence, all of the experiments reported in this section were conducted on the NC data unless stated otherwise.

		German	English
Training (Adaptation $\mathcal{A}$)	Sentences	100k	
	Run. words	2.5M	2.4M
	Vocabulary	102.6k	47.2k
Test 2009	Sentences	2525	
	Run. words	62.7k	65.6k
	OoV. words	3352	1683

Table 4.1: Main figures of the News-Commentary corpus. *OoV* stands for Out of Vocabulary. k/M stands for thousands/millions of elements.

		French	English
Training (Adaptation $\mathcal{A}$)	Sentences	47.5k	
	Run. words	792.9k	747.2k
	Vocabulary	31.7k	24.6k
Test	Sentences	641	
	Run. words	12.8k	12.6k
	OoV. words	954	427

Table 4.2: Main figures of the TED corpus. *OoV* stands for Out of Vocabulary. k/M stands for thousands/millions of elements.

4.8.2 Machine translation evaluation measures

For the purpose of computing the best hypothesis $\boldsymbol{y}^*$ as described in Equation 4.23, TER will be used. Although BLEU is slightly more popular in the SMT community, BLEU is only well defined on the corpus level, but not on the sentence level (see Section 1.2.2). Hence, it is not well suited for our purposes since the complete set of n-best candidates provided by the decoder can score zero. For coherence reasons, results will be reported with TER.

In the case of online adaptation, translation quality will be measured before adaptation takes place, i.e., first the system will propose a hypothesis, then the translation quality of that hypothesis is evaluated, and finally the adaptation procedure is activated. This implies that the final translation quality is the average over the complete test set, although the system was not adapted at all when translating the first sentences.

4.8.3 Batch adaptation results

In the first place, the effect of BPA in a batch setup was studied, i.e., in a scenario where there is an adaptation set available beforehand. In this context, all of the sampling algorithms described in Section 4.6 can be applied. The experiments reported in the following were conducted by using the Europarl training data as training set $\mathcal{T}$ and the Europarl development data for estimating the initial set of weights $\boldsymbol{\lambda}_{\mathcal{T}} \equiv \boldsymbol{\lambda}_{\xi}$ (see Table 1.1). The baseline system reported refers to the non-adapted system, i.e., using $\boldsymbol{\lambda}_{\mathcal{T}} \equiv \boldsymbol{\lambda}_{\xi}$ as weight vector within the decoder to obtain the final translations. The adaptation set $\mathcal{A}$ was extracted from the News-Commentary or TED training data at random, and this extraction was performed 10 times,

so that each one of the points in the plots presented in this chapter constitutes the average of these 10 repetitions. Finally, the final test set used for evaluation purposes was the 2009 test set (see Table 4.1) in the case of the News-Commentary corpus, and the Test set in the case of the TED corpus.

To synthesise the experimental setup, the different SMT systems compared in this section when adapting $\boldsymbol{\lambda}$ are:

- Baseline system: Phrase-pairs extracted from the Europarl training corpus (i.e., $\boldsymbol{h}$ estimated on the Europarl training data). Log-linear weights $\boldsymbol{\lambda}$ estimated on the development partition of the Europarl corpus, $\boldsymbol{\lambda}_{\mathcal{T}} \equiv \boldsymbol{\lambda}_{\xi}$
- BPA: Initial setup identical to the baseline system. Then, adaptation samples $\mathcal{A}$ were randomly extracted from the training partitions of the in-domain corpora (i.e., NC or TED). The set $\boldsymbol{\lambda}$ estimated on the Europarl development data is used as $\boldsymbol{\lambda}_{\mathcal{T}}$ within the parameter prior $p(\boldsymbol{\lambda} \mid \mathcal{T})$ in all the experiments concerning the adaptation of $\boldsymbol{\lambda}$.
- MERT: Initial setup identical to the baseline system. Then, the adaptation samples $\mathcal{A}$ described above were used for estimating a new set of log-linear weights by means of MERT.
- MERT+: Initial setup identical to the baseline system. Then, both $\mathcal{A}$ and the Europarl development set were used for estimating a new set of $\boldsymbol{\lambda}$.

The `MERT` and `MERT+` settings will be used in the last part of this section, when comparing the performance of the BPA systems.

The first experiments conducted were performed by adapting the scaling factors $\boldsymbol{\lambda}$ and with the purpose of analysing the effect of the different parameters involved in the heuristic sampling strategy, such as the leveraging factor δ, the prior variance $\sigma_{\mathcal{T}}$, or the size of $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$. With the same purpose, additional experiments were performed for the Viterbi, Gaussian and MCMC sampling strategies. Since feature function adaptation is much more costly than adapting the scaling factors $\boldsymbol{\lambda}$, most of the experiments reported involve adapting $\boldsymbol{\lambda}$, although some experiments adapting $\boldsymbol{h}$ are also reported. Adapting $\boldsymbol{\lambda}$ by means of BPA is compared with using the adaptation set $\mathcal{A}$ as development set for re-estimating $\boldsymbol{\lambda}$ from scratch by means of MERT, and also with re-estimating $\boldsymbol{\lambda}$ by using both the adaptation data $\mathcal{A}$ and the development set.

Heuristic sampling

Effect of different values of the leveraging factor δ Results for this kind of sampling are shown in Figure 4.2, for different values of the δ leveraging factor and with increasing number of adaptation samples.

On the one hand, the plot on the left side displays translation quality, as measured by TER. As shown, BPA is able to improve over the unadapted system from the very beginning. Regarding the effect of δ, the results show that this parameter leveraging factor has an important role in the confidence interval sizes, which is why increasing δ leads to smoother adaptation curves. In addition, smaller values of δ lead to a slight degradation in translation quality when the amount of adaptation samples becomes larger, and also present slightly more noisy

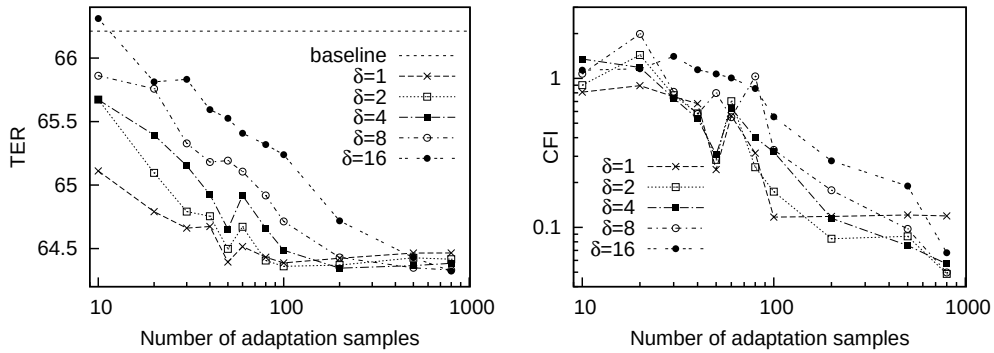


Figure 4.2: Batch adaptation for different values of delta. News-Commentary corpus considered. In these plots, the size of the n-best list was fixed to 200.

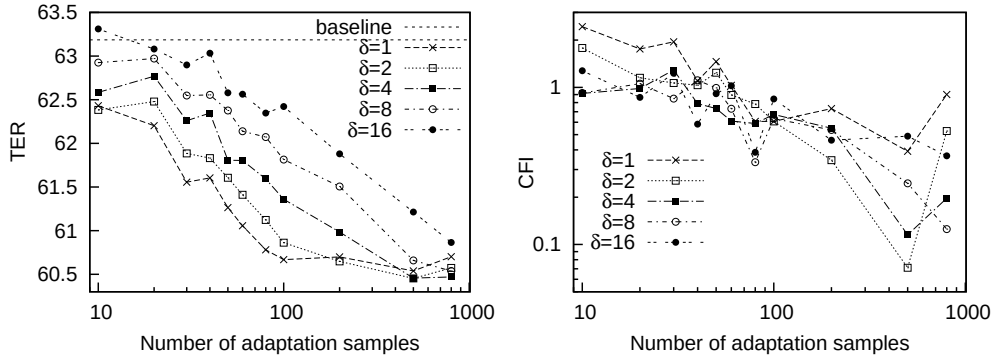


Figure 4.3: Batch adaptation for different values of delta for the TED corpus. In these plots, the size of the n-best list was fixed to 200.

curves, i.e., with larger confidence intervals. The reason for this can be explained by looking at Equation 4.16. Since $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})$ is in practise implemented as a product of probabilities, the more adaptation samples the smaller becomes $p(\mathcal{A} \mid \boldsymbol{\theta}; \mathcal{T})$, and a higher value of δ is needed to compensate this fact. Although larger values of δ do not suffer the problem described, they yield smaller improvements in terms of translation quality for smaller amount of samples. This suggests the need of a δ which depends on the size of the adaptation sample. Despite the fact that the differences between different δ values observed in Figure 4.2 for larger adaptation set sizes are very small, and are in fact only statistically significant in some cases, such differences were found to be coherent in other language pairs and other corpora (see Figure 4.3 for results on the TED corpus). Values for δ smaller than 1 were also analysed, although the resulting curves ended up always between the ones corresponding to $\delta = 1$ and $\delta = 16$, but without displaying a clear behaviour. This result is actually quite logical, since

values of δ smaller than 1 do not have much sense from a theoretical point of view either.

Effect of the prior distribution variance $\sigma_{\mathcal{T}}$ Another (meta-) parameter that needs to be fixed empirically is the variance of the normal distribution of the model parameters, i.e., $p(\boldsymbol{\theta} \mid \mathcal{T}) \sim \mathcal{N}(\boldsymbol{\lambda}_{\mathcal{T}}, I \cdot \sigma_{\mathcal{T}})$. For doing this, $\delta = 4$ was chosen, according to the experiments detailed above and given that it appears to be the value that presents a good compromise in quality for small and big adaptation set sizes and in addition presents a more smooth behaviour than the curves with smaller values of δ. The result of considering different values for $\sigma_{\mathcal{T}}$ is shown in Figure 4.4. The effect of $\sigma_{\mathcal{T}}$ in the performance achieved by BPA is very important, since low values of $\sigma_{\mathcal{T}}$ lead to low variability and no adaptation takes place. On the other hand, too high values of $\sigma_{\mathcal{T}}$ may yield too abrupt changes, leading to over-trained adaptation curves and larger confidence intervals. Confidence intervals did not seem to present important changes when varying $\sigma_{\mathcal{T}}$, and are hence omitted here for clarity reasons. However, $\sigma_{\mathcal{T}} = 0.1$ did seem to yield slightly smaller confidence intervals than $\sigma_{\mathcal{T}} = 0.01$, which is the reason why the rest of the experiments in this section were performed with $\sigma_{\mathcal{T}} = 0.1$. For $\sigma_{\mathcal{T}} \geq 1$, the adaptation curves were practically indistinguishable.

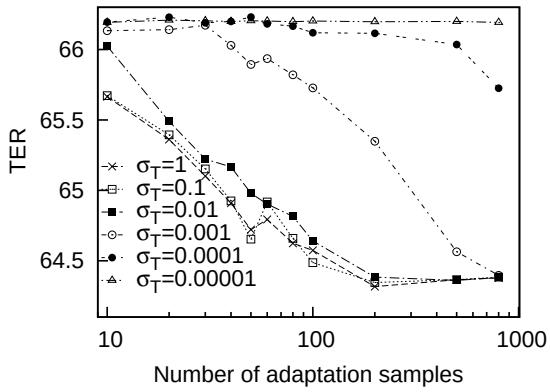


Figure 4.4: Translation quality for different variances $\sigma_{\mathcal{T}}$. In this case, δ was set to 4 and the size of $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ was set to 200.

Considering different n-best list sizes As said in Section 4.7, the BPA implementation used in the present work approximates the summation $\sum_{\boldsymbol{y}'}$ as the sum over a given n-best list. Moreover, the best hypothesis that the system is able to deliver is also selected from such an n-best list. For these two reasons, it is also interesting to study the behaviour of BPA when incrementing such n-best list, and this was done once δ and $\sigma_{\mathcal{T}}$ had been fixed empirically. In order to avoid an overwhelming amount of results, only those results obtained when considering 100 adaptation samples are displayed in Figure 4.5. As it can be seen, TER drops quite monotonically for all δ values, until about 800, where it starts to stabilise. We consider that this is also an interesting result. When increasing the n-best list size, it is probable that the hypothesis $\boldsymbol{y}^*$ is chosen from a deeper position in such list. Although this

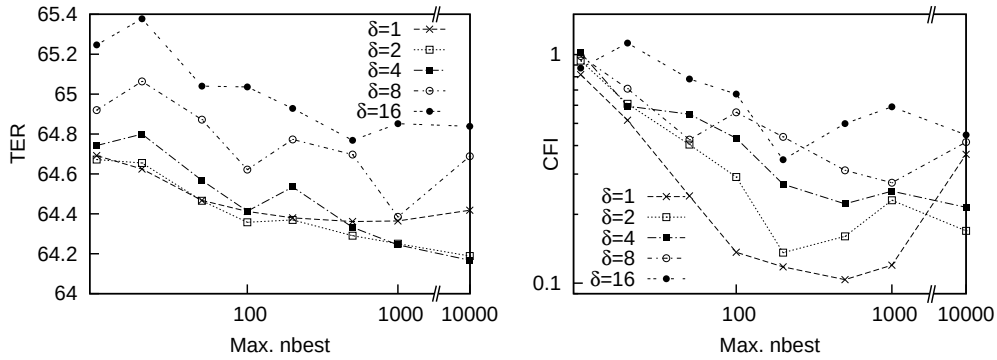


Figure 4.5: Batch adaptation with heuristic sampling for different values of δ and n-best. The left plot displays translation quality as measured by TER and the right plot displays confidence interval sizes. The size of the adaptation data was fixed to 100 sentences. Note that, for clarity reasons, the x-axis is *broken*, meaning that the distance between the 1000 and 10000 ticks is actually altered.

sounds reasonable, it could also be possible that deepening into the n-best list would yield degenerate values of TER (Martínez-Gómez et al., 2011), hence leading to an over-trained system. However, this does not seem to be the case with BPA.

Effect of increasing the amount of sampled parameters Finally, we also studied the effect of varying the number of sampled parameters $|\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})|$. Theoretically, increasing the size of this set size should only lead to more stable results, but should not have any effect in terms of translation quality: whenever it can be assumed that $\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})$ is a good representative of the true distribution of the model parameters $\boldsymbol{\theta}$, increasing the number of sampled parameters should only provide more robustness. As expected, (average) translation quality was not affected by the size of $\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})$, and the curves obtained were almost identical. For this reason, only the confidence interval sizes are reported here. Such results are shown in Figure 4.6, with the amount of sampled weights $|\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})|$ being represented in the plot by `nw`. The results show that the more sampled $\boldsymbol{\lambda}$, the more stable the results appear to be. However, when increasing $|\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})|$ from 1000 to 2000, the improvements in stability are already very scarce, and might not be worth the computational overhead.

Viterbi approach

The Viterbi approach described in Section 4.6.4 was also analysed, and the results obtained are shown in Figure 4.7, for different values of δ. When comparing this set of plots with Figure 4.2, it is interesting to realise that the effect of the Viterbi approach is that the leveraging factor δ has practically no effect. This is true both when the amount of adaptation samples is low, but also when the amount of adaptation samples increases. On the one hand, this is a desirable behaviour, since it drops the necessity of using δ when dealing with small adaptation

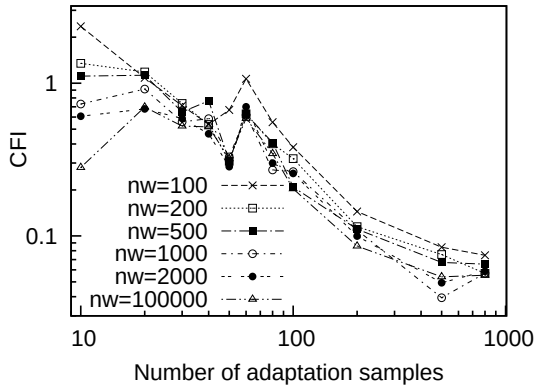


Figure 4.6: Effect of increasing the size of $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$, i.e., $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})|$ denoted by nw in the plot, on the size of the confidence intervals. δ was set to 4, and the size of the n-best list to 200.

sets. On the other hand, however, it means that for larger adaptation sets δ does not compensate the problem described in Section 4.6.1 and all adaptation curves seem to re-bounce after about 100 adaptation samples have been seen. The fact that all curves present a very similar behaviour may be due to the own nature of this sampling strategy: since $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ is restrained to one single-best $\boldsymbol{\lambda}$, chosen according to the distribution to be sampled, the results are bound to be very similar.

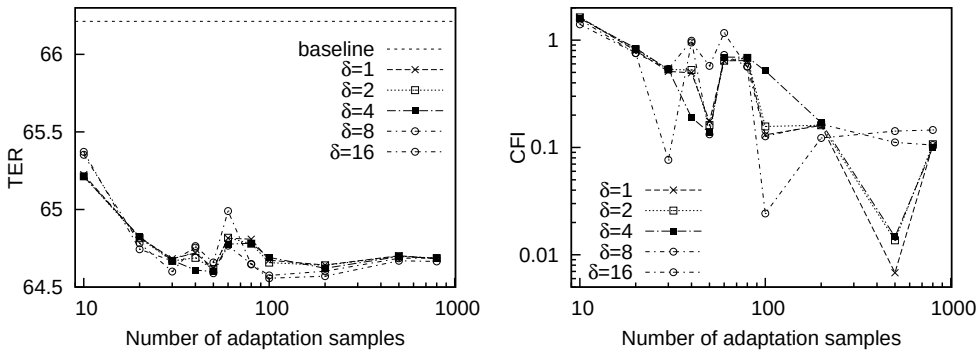


Figure 4.7: Batch adaptation with Viterbi sampling and different amount of adaptation set sizes. The size of the n-best list was fixed to 200 and $\sigma_{\mathcal{T}} = 0.1$.

Gaussian sampling

The next set of experiments involved sampling only according to the Gaussian prior. Results for different $\sigma_{\mathcal{T}}$ values are shown in Figure 4.8. Different $\sigma_{\mathcal{T}}$ values did not seem to affect the final translation quality, and the adaptation curves present almost the same shape. As shown, this sampling strategy performs almost as well as the heuristic approach until about 80 adaptation samples. At that point, the curves start to bounce back in a more chaotic fashion than in the case of heuristic sampling. Most likely, this is due to the larger confidence intervals entailed by sampling from Gaussian prior, when compared to those obtained with the heuristic sampling, as shown in the right part of the figure. The reason for this might be that Gaussian sampling introduces less variability than the heuristic sampling strategy because of their own nature: Gaussian sampling obtains many $\boldsymbol{\lambda}$-samples from the close neighbourhood of $\boldsymbol{\lambda}_{\mathcal{T}}$, because of the shape of the Gaussian distribution, while the heuristic sampling strategy is able to obtain more different λ samples. Hence, the hypothesis provided as output in the case of Gaussian sampling has been chosen by observing less variability in $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$, and is thus less robust. Of course, having more variability in $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ while ignoring completely the distribution being sampled is not beneficial as such, but given that the true distribution being sampled contains the probability of the adaptation data $\mathcal{A}$, and such probability is ignored by both the heuristic and Gaussian strategies, increasing variability may be, to a certain extent, the best way to include into $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ samples which are actually near the peak of the true distribution that should sampled.

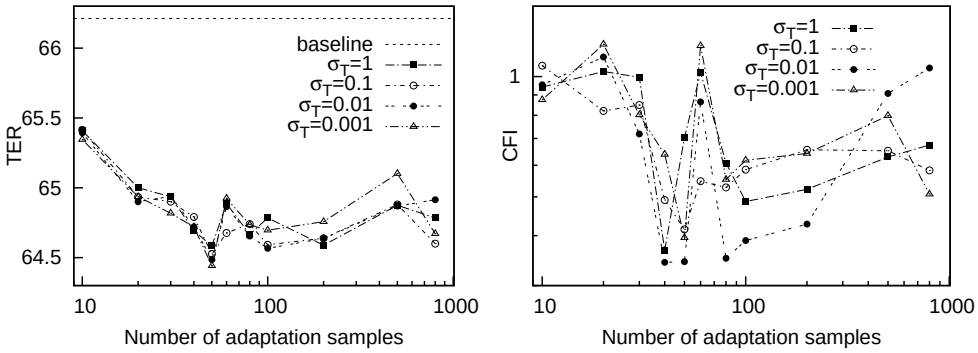


Figure 4.8: Gaussian sampling strategy for batch adaptation of $\boldsymbol{\lambda}$.

MCMC sampling

As for MCMC sampling, the first experiments were conducted in order to establish appropriate values for prior and proposal distribution variances ($\sigma_{\mathcal{T}}$ and σ_o, respectively) and the interaction thereof. As for the case of $\sigma_{\mathcal{T}}$ in heuristic sampling, $\sigma_{\mathcal{T}}$ and σ_o have a very important role in MCMC. As explained in Section 4.6.3, on the one hand, small values of σ_o lead to slow mixing chains and no adaptation would take place, but on the other hand too high

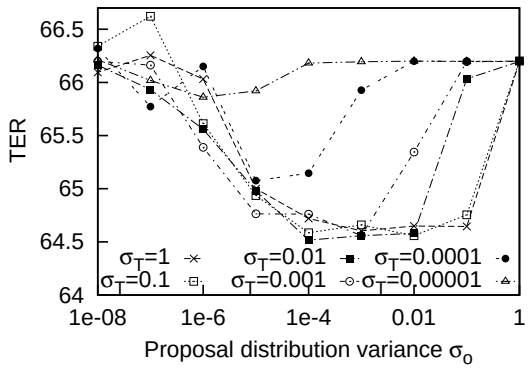


Figure 4.9: Translation quality for different values of $\sigma_{\mathcal{T}}$ and σ_o within the MCMC procedure for BPA. $\delta = 1$ and size of n-best set to 200.

values lead to noisy chains that never converge and hence do not yield appropriate results. Furthermore, σ_o is tightly related to $\sigma_{\mathcal{T}}$, since they both control how much variation is introduced into the predictive distribution of BPA. Their interaction is shown in Figure 4.9. This plot displays the translation quality that can be achieved for a given $\sigma_{\mathcal{T}}$, when varying the proposal distribution variance σ_o. As expected, $\sigma_{\mathcal{T}}$ and σ_o seem to be very closely related, and the best values for σ_o depend on the prior distribution variance, with all curves presenting an optimum at $\sigma_o = 0.1 \cdot \sigma_{\mathcal{T}}$. Considering $\sigma_o > \sigma_{\mathcal{T}}$ seems to lead to systems where no adaptation takes place, and all curves remain steadily at the baseline translation quality whenever the proposal distribution variance is higher than the prior variance. As for the case of heuristic sampling, adequate values for $\sigma_{\mathcal{T}}$ seem to be 1 or 0.1. Again, in the rest of the experiments within this Section, $\sigma_{\mathcal{T}}$ was set to 0.1 for having slightly smaller confidence intervals, and hence $\sigma_o = 0.01$. Confidence intervals are omitted in this case because they were very similar, except for the cases were no adaptation takes place, where confidence interval sizes were very near to 0.

Another aspect that needs to be taken into account when working with MCMC is the length of the burn-in phase. As Figure 4.10 shows, this aspect of the MCMC chain does have a slight effect on the stability of the resulting system, although it fades away when increasing the size of $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$. This was quite expected, since increasing the length of the Markov chain implies that the initial noise it might contain is smoothed by the rest of the chain. However, when observing the plot, it does seem that an appropriate burn-in phase should contain between 500 and 1000 samples, although the differences observed are so scarce and incoherent that no final conclusion could be drawn. Nevertheless, after observing this plot, the length of the burn-in phase was set to 500 in the rest of the experiments of this chapter that involve MCMC.

As for the effect of considering different $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ sizes, i.e., different MCMC chain lengths, the results of such experimentation are presented in Figure 4.11. In the left plot, translation quality is shown, whereas the right plot displays the size of the confidence intervals in the

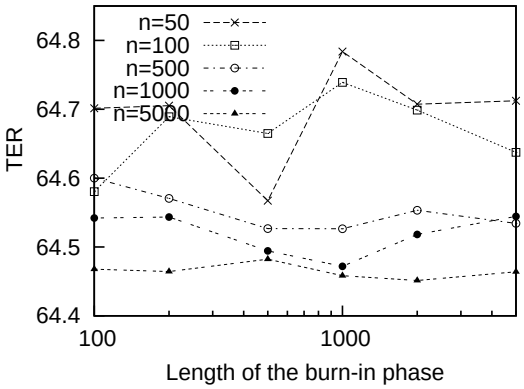


Figure 4.10: Effect of considering different burn-in durations, when varying the MCMC chain length. The number of sampled weights after the burn-in phase (i.e., $|\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})|$) is denoted by n. The number of adaptation samples was set to 100. $\sigma_\mathcal{T} = 0.1$ and $\sigma_o = 0.01$.

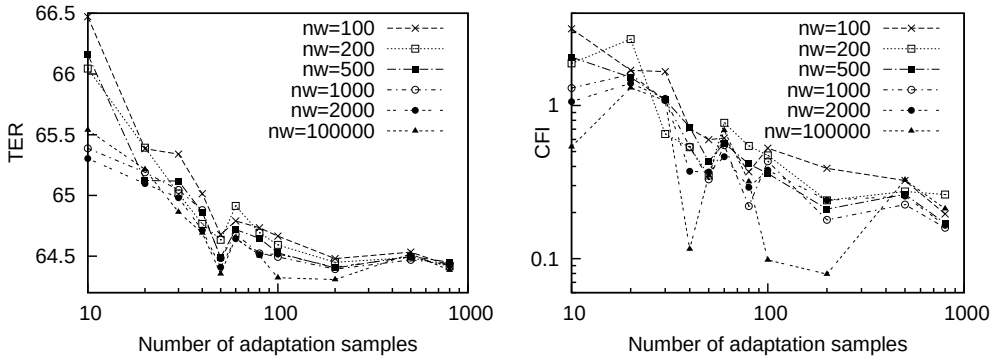


Figure 4.11: Effect on translation quality and confidence interval sizes of considering different $\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})$ sizes, denoted by nw in the plot. Burn-in duration was set to 200. $\sigma_\mathcal{T} = 0.1$ and $\sigma_o = 0.01$.

logarithmic scale. Although it might seem that $\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})$ size is a critical factor when applying MCMC in BPA, such conclusion is not completely true. Taking a closer look at the confidence intervals, these were as large as 3 TER points when considering only 10 samples of $\boldsymbol{\theta}_\mathcal{T}$. Hence, differences observed in terms of translation quality are not significant. What is significant, however, is that stability in BPA is achieved by increasing the number of observations of $\boldsymbol{\theta}_\mathcal{T}$ that approximate the integral in Equation 4.9. As for the case of the heuristic method above, the difference in stability between performing 1000 or 2000 sampling steps may not be worth the computational overhead, since at that point the curves present almost the same shape.

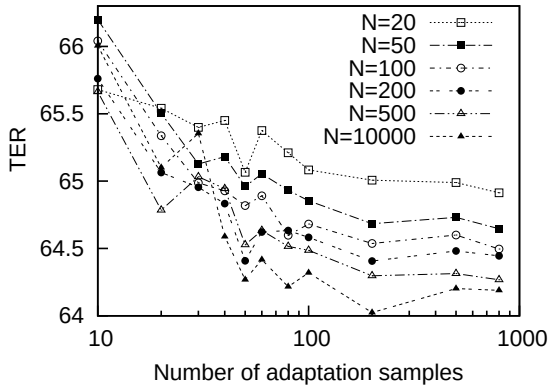


Figure 4.12: Translation quality for batch adaptation with MCMC sampling and different sizes of n-best, represented as `N` in the plot. $\sigma_{\mathcal{T}} = 0.1$ and $\sigma_o = 0.01$.

Lastly, and as done for the case of the heuristic sampling above, we also analysed the effect of varying the size of the n-best list considered. Such results are shown in Figure 4.12. As in the case of heuristic sampling, BPA is able to cope well with additional input information, and additional hypotheses in the n-best list imply that BPA is able to select better hypothesis without incurring into over-trained solutions.

Comparison between BPA and parameter re-estimation

In addition to results with BPA and for comparison purposes, experiments using $\mathcal{A}$ as development set for performing a full re-estimation of $\boldsymbol{\lambda}$ with MERT were also conducted. However, it could be argued that such setup is not a fair comparison, since BPA also makes use of the information obtained in the training phase, such information being contained within the prior over the parameters. For this reason, we also provide results obtained by re-estimating $\boldsymbol{\lambda}$ on a development set built of the original development set used for the initial estimation, and the adaptation data, both concatenated, i.e., $\mathcal{D} \cup \mathcal{A}$. This setup will be referred to as `MERT+`. Nevertheless, note that such baselines are not really a fair comparison. On the one hand, because they are both by far much more costly than BPA, since re-estimating the parameters from scratch takes several hours or even days, whereas the BPA implementation takes only a couple of minutes. On the other hand, because the MERT procedure involves several translation steps, each of which re-computes the n-best list and hence has better chances to obtain better hypotheses.

Results of such comparison can be seen in Figure 4.13, where only the heuristic and MCMC sampling strategies are reported in order to avoid clogging the plots. It can be seen that BPA is able to provide better results than re-estimating $\boldsymbol{\lambda}$ from scratch for small sizes of $\mathcal{A}$. If such re-estimation is carried out by using only $\mathcal{A}$, it comes to a point where it performs better than BPA. However, re-estimating $\boldsymbol{\lambda}$ by using both $\mathcal{A}$ and the Europarl development data ($\mathcal{D}$) provides significantly worse results.

On the other hand, the `MERT` setup displays a rather chaotic curve, which can be explained

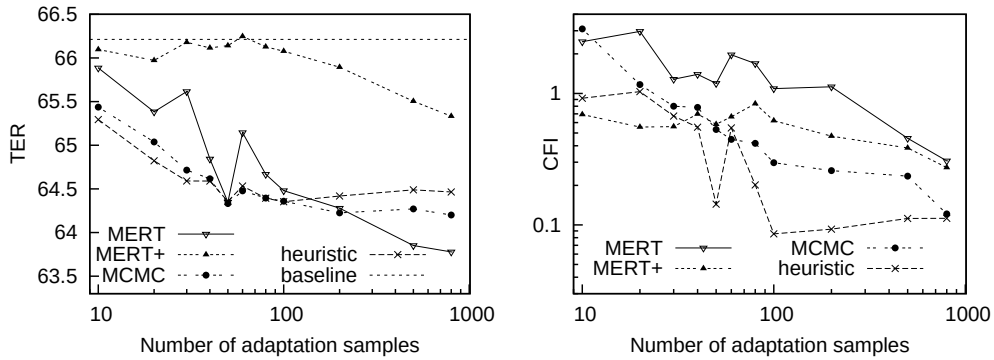


Figure 4.13: Translation quality, as measured by TER, obtained when comparing performing a full re-estimation of $\boldsymbol{\lambda}$ by means of MERT, and when using the same adaptation data as adaptation set within BPA. News-Commentary corpus considered.

when looking at the plot on the right, which depicts the size of the confidence interval sizes in logarithmic scale. For small sizes of $\mathcal{A}$, such intervals are relatively large for the case of `MERT`, as large as 3 TER points. However, in the case of BPA they are much smaller, as small as 0.6 even for as few as 10 adaptation samples. In contrast, `MERT+` yields very small confidence interval sizes, but, as seen previously, is not able to provide better performance than BPA.

Regarding the performance of the MCMC sampling strategy when compared to the heuristic sampling, the experimental results in Figure 4.13 show that the heuristic strategy is able to yield better results in terms of translation quality than the MCMC strategy, until about 100 adaptation samples, which is the point where the normalisation problem described in Section 4.6.1 starts to appear. Nevertheless, it is at that point where the advantages provided by BPA start to fade. In addition, the heuristic strategy provides smaller confidence interval sizes, and, furthermore, it is much cheaper in terms of computational resources. Hence, it can be stated that the heuristic strategy is the one that yields the best results, when applying BPA to SMT.

Additional experiments comparing BPA with both sampling strategies and the MERT baselines were performed on the TED corpus. The meta-parameters in BPA were set according to the experiments performed previously on the News-Commentary corpus. Such experiments are shown in Figure 4.14, for the case of TER, and in Figure 4.15 for the case of BLEU. In this case, the conclusions to be drawn are similar to those obtained from the News-Commentary corpus, although in this case both MERT setups behave slightly worse than in the previous case. More specifically, the `MERT` setup presents very high confidence intervals when the amount of adaptation samples is low, and the `MERT+` setup does not achieve to perform significantly better than the baseline setup in any case. Meanwhile, both BPA settings are able to improve performance from the very beginning, improving the baseline by more than 2 TER (1-2 BLEU) points with as few as 50 adaptation samples. In terms of BLEU, the BPA approaches seem to behave in a slightly less predictable fashion. However, this is

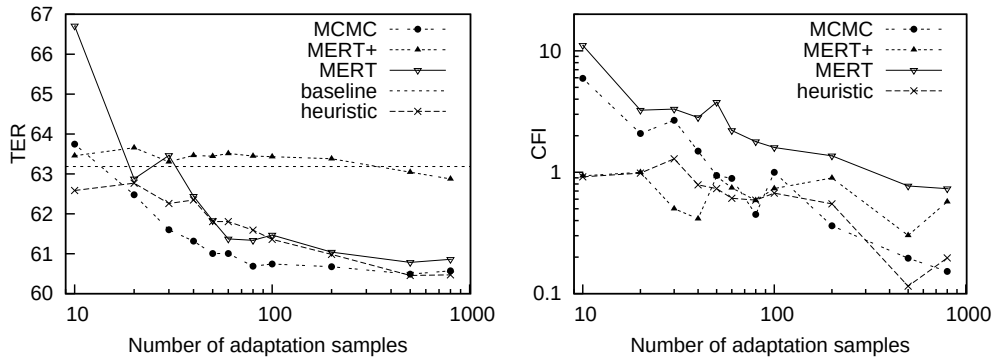


Figure 4.14: Translation quality, as measured by TER, obtained when comparing performing a full re-estimation of $\boldsymbol{\lambda}$ by means of MERT, and when using the same adaptation data as adaptation set within BPA. TED corpus considered.

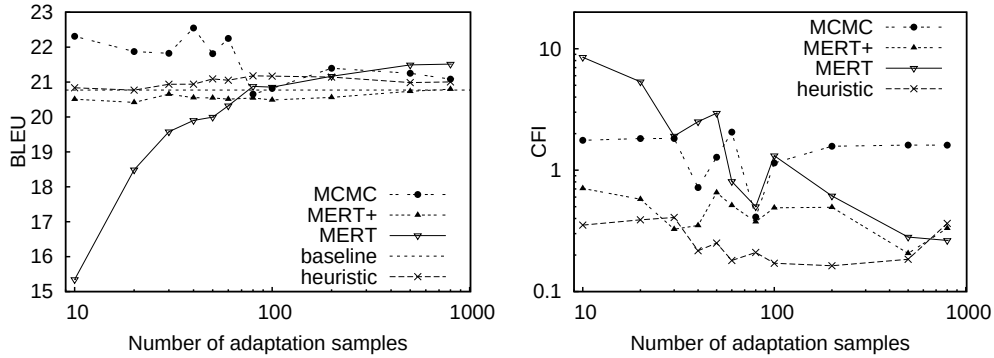


Figure 4.15: Translation quality, as measured by BLEU, obtained when comparing performing a full re-estimation of $\boldsymbol{\lambda}$ by means of MERT, and when using the same adaptation data as adaptation set within BPA.TED corpus considered.

actually expected, since the best possible hypothesis $\boldsymbol{y}^*$ is selected according to TER.

With the purpose of getting some insight about where the improvements come from, we analysed the n-gram precision and the brevity penalty implemented within BLEU. For a certain n, n-gram precision is computed as the number of n-grams that match between the candidate hypotheses and the references, normalised by the total amount of n-grams that constitute the references. The brevity penalty is defined as $\min(1, r)$, being r the ratio between hypothesis and reference lengths, and gives an insight about how well the SMT system is predicting the length of the reference translations. By analysing n-gram precision and brevity penalty, the purpose is to elucidate whether the improvements achieved are due to a better lexical choice of the translation units, or rather due to a better prediction of reference

		10 adaptation samples				100 adaptation samples			
	baseline	Heur.	MCMC	MERT	MERT+	Heur.	MCMC	MERT	MERT+
BLEU	16.7	16.4	15.8	14.2	16.9	16.1	16.0	16.2	16.8
1-gram	54.9	55.5	56.0	56.2	54.9	56.7	56.8	56.2	55.0
2-gram	23.5	23.6	23.5	22.3	23.5	24.2	24.2	23.9	23.4
3-gram	11.5	11.6	11.3	10.5	11.6	11.8	11.9	11.8	11.5
4-gram	6.0	6.0	5.8	5.2	6.0	6.1	6.1	6.1	6.0
brev. pen.	0.97	0.95	0.92	0.88	0.98	0.91	0.90	0.92	0.97

Table 4.3: Analysis of n-gram precision and brevity penalty for 10 and 100 adaptation samples, considering heuristic and MCMC sampling within BPA and MERT and MERT+ strategies (i.e., including the adaptation data only, or the adaptation and development data for re-estimating $\boldsymbol{\lambda}$. NC corpus considered.

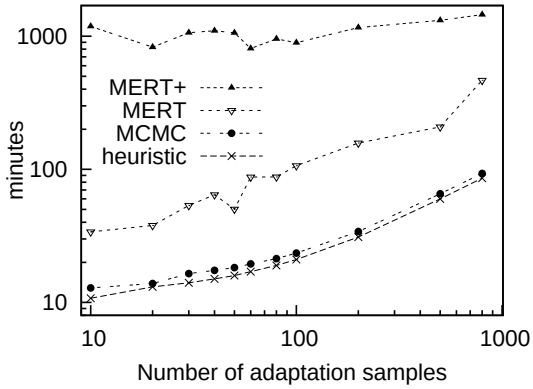


Figure 4.16: Time in minutes consumed by the different adaptation approaches compared. In the case of BPA, 2000 samples for $\boldsymbol{\lambda}$ were obtained, and the size of the n-best list was set to 200. Note that both axes are shown in logarithmic scale. News-Commentary corpus considered.

length. These results are shown in Table 4.3 when using 10 and 100 adaptation samples for the case of the News-Commentary 2009 test set. In this table, it is interesting to see that both BPA approaches and `MERT` are able to yield higher n-gram precision rates than the baseline, and even than the `MERT+` setup, but are severely penalised by the brevity penalty, leading to significantly lower BLEU scores than the baseline. This was actually expected, since the TER score considered within BPA does not include the brevity penalty. However, the fact that n-gram precision is higher leads to the conclusion that the improvements obtained over the baseline are due to a better lexical choice of the phrases involved in the translation process, and not to a side-effect of adjusting the output sentence length.

One last word regarding this comparison involves computational time. Figure 4.16 reports the time consumed by each one of the approaches reported in Figure 4.13. In the case of the two BPA strategies, the amount of sampled weights was 2000, i.e., the most costly and stable

experiments involving BPA conducted in this section. Regarding the time taken by both BPA approaches, it must be noted that both include the time consumed for generating the n-best lists for the adaptation data and the sentence-level TER counts. In fact, the time taken only by the BPA implementation ranges from 10 minutes up to 20, depending on the amount of adaptation samples considered, but computing the sentence level TER counts gets specially costly when longer sentences are involved. As the plot shows, both BPA implementations take much less time than the MERT alternatives, being the heuristic BPA alternative the fastest one in a consistent manner. Note that, in Figure 4.16, the y-axis is plotted in logarithmic scale, which implies that the BPA implementations are about one order of magnitude faster than the `MERT` alternative, and two orders of magnitude faster than `MERT+`.

Feature function adaptation

Preliminary experiments conducting Bayesian predictive adaptation of the model features $\boldsymbol{h}$ were also performed. However, given the extremely high computational cost involved, only a small number of these experiments were performed. Specifically, in the case of the NC 2009 test set the adapted system achieved a TER score of 66.0, compared to 66.2 of the baseline system. In the case of the TED test set, the adapted system achieved a TER score of 63.0, compared to 63.2 of the baseline system. This (minor) improvement was achieved by setting $\delta = 32$ and with 2000 adaptation samples. However, these experiments are extremely costly from a computational perspective. Even when combining all the features defined at the local phrase level, as described in Section 4.3, and performing the approximations described in Section 4.7, re-scoring the translation hypotheses obtained when translating the test set takes about one week in a single-threaded implementation with only 1000 repetitions of the heuristic sampling algorithm described in Section 4.6. For these reasons, and although there seems to be some potential in the adaptation of the feature functions $\boldsymbol{h}$, no further experiments in this direction were performed.

4.8.4 Online adaptation results

In the previous section it has been shown that MCMC has a more reliable behaviour in a batch adaptation setup than the heuristic algorithm. Nevertheless, when confronting an online adaptation problem, time constraints imply that MCMC is not applicable, since $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$ would need to be redrawn for each new adaptation sample seen by the system. Alternatively, sampling from the Gaussian prior seems to be slightly more unstable than the heuristic sampling strategy. For these reasons, only experiments with the heuristic algorithm were performed for online adaptation, and only for the adaptation of the log-linear weights $\boldsymbol{\lambda}$, since adapting $\boldsymbol{h}$ proved to be too expensive for an online BPA implementation.

The result of applying BPA in an online setting can be seen in Figure 4.17. In this figure, the x-axis is the amount of trailing samples considered, i.e., the number of trailing sentences that are included into the set $\mathcal{A}_t$ described in Section 4.4. This figure only includes the resulting translation quality because the confidence interval sizes did not seem to vary much with δ, as was the case with batch adaptation. It is interesting to point out that the translation quality curves seem to present a minimum at about 100 adaptation samples, and adding further trailing sentences into $\mathcal{A}_t$ seems to actually produce a significant degradation in the final

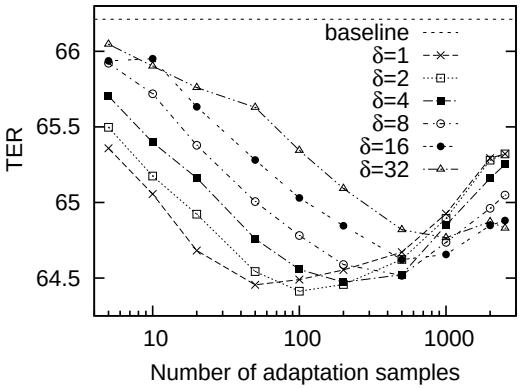


Figure 4.17: Effect of different δ leveraging factors in online adaptation. The size of the n-best list was fixed to 200. $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 2000$ and $\sigma_{\mathcal{T}} = 0.1$.

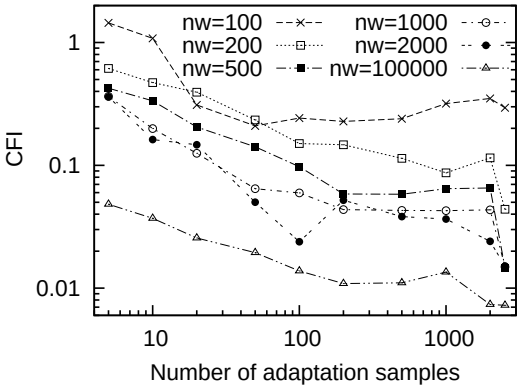


Figure 4.18: Confidence interval sizes for different sizes of $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$, denoted by `nw` in the plot.

translation quality achieved. This seems to point towards the possibility that there is a certain locality in the weights.

As for the amount of sampled weights, $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})|$, varying this value did not appear to produce any change in terms of translation quality. However, this was so only when considering the average of all the 10 repetitions performed, since the size of the confidence intervals did present interesting changes. Such confidence intervals are shown in Figure 4.18. As expected after the experiments conducted with batch adaptation, the size of the confidence intervals drops significantly when increasing the size of $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$, yielding very small confidence intervals when $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 100000$. Nevertheless, each one experiment with $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 100000$ takes about 20 hours, when compared to several minutes in the case of $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 2000$.

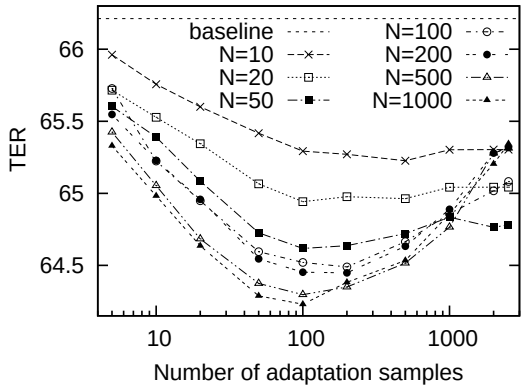


Figure 4.19: Translation quality when considering increasing n-best size. $\delta = 2$, $\sigma_{\mathcal{T}} = 0.1$ and $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 2000$.

For this reason, the additional decrease in confidence interval sizes might not be worth beyond $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 2000$. In addition, note that the size of these confidence intervals cannot be compared directly with the size of the confidence intervals shown when applying BPA in a batch setup, since the experimentation in the batch setup entailed obtaining a new adaptation sample at random for each point in the plot. This is not the case when dealing with online adaptation because re-drawing the adaptation data is not possible because the adaptation data $\mathcal{A}_t$ is fixed to be the last sentences observed in the current test set being translated.

The effect of increasing the size N of the n-best list was also analysed. Results for $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 1000$ and $|\mathcal{A}_t| = 100$ are shown in Figure 4.19. As was expected, the translation quality provided by including the sliding window $\mathcal{A}_t$ improves when increasing the amount of n-best considered. However, this improvement seems to decay gradually, and increasing N from 500 to 1000 already yields scarce improvements.

Finally, in Fig. 4.20 the results of varying $\sigma_{\mathcal{T}}$ of Eq. 4.10 are shown. The translation quality delivered by the Bayesian sliding window is, in the worst case, the same as the baseline system. For lower values of $\sigma_{\mathcal{T}}$, the sliding window has no effect at all until about 100 samples. The optimal value for $\sigma_{\mathcal{T}}$ seems to be 1 or 0.1. For all experiments above, $\sigma_{\mathcal{T}} = 0.1$ was chosen due to having slightly smaller confidence intervals and because this was also the value chosen in the case of batch adaptation.

4.8.5 Bayesian adaptation for system stabilisation

Lastly, experiments concerning the use of BPA for system stabilisation purposes, as described in Section 4.5, were also conducted. For doing this, a low-resource environment was simulated by randomly selecting a (small) training set $\mathcal{T}$ from the News-Commentary training data. In addition, a random development set $\mathcal{D}$ of 100 sentences was also extracted from the remainder News-Commentary training data, ensuring that $\mathcal{D}$ and $\mathcal{T}$ are fully disjoint, i.e., $\mathcal{D} \cap \mathcal{T} = \emptyset$. Note that this development set is not the same one referred to in the previous

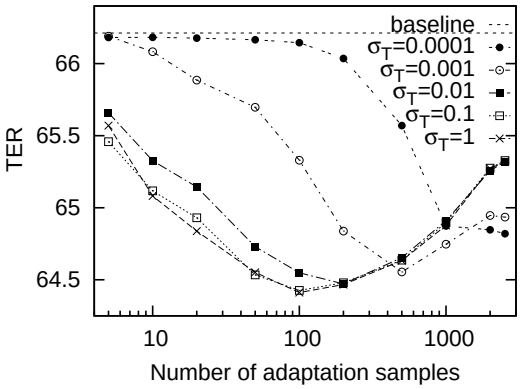


Figure 4.20: Effect of varying $\sigma_{\mathcal{T}}$ within $\mathcal{N}(\boldsymbol{\lambda}_{\mathcal{T}}, I \cdot \sigma_{\mathcal{T}})$. The size of the n-best list was set to 200. $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 2000$ and $\delta = 2$.

experiments, since in this case $\mathcal{D}$ was obtained from the NC data, and in the previous experiments it was a fixed set belonging to the Europarl data. Then, the random training set $\mathcal{T}$ was used for phrase extraction and building the phrase-table, while the random development set $\mathcal{D}$ was used within MERT for estimating the corresponding set of weights $\boldsymbol{\lambda}_{\mathcal{D}}$. This being done, two different approaches were used for BPA. In the first option, the set of weights estimated with MERT was used as mean vector within the parameter prior in BPA, i.e $\boldsymbol{\lambda}_{\mathcal{T}} \equiv \boldsymbol{\lambda}_{\mathcal{D}}$ (first option described in Section 4.5). In the second option, the set of weights estimated for Europarl by means of MERT was used as parameter prior, i.e., $\boldsymbol{\lambda}_{\mathcal{T}} \equiv \boldsymbol{\lambda}_{\xi}$ (second option in Section 4.5). Finally, the test set used for the final evaluation was the same as in the previous experiments. The results of this setup are shown in Figure 4.21. As shown, all the alternatives present decreasing TER scores when adding more training data, as expected. However, the two BPA approaches perform slightly better in average than the MERT approach. In addition, taking a look at the confidence interval sizes reveals an interesting result: the confidence intervals are smaller when using BPA, which actually means that applying BPA as a post-processing step does actually provide more stable results. Finally, it can also be seen that using $\boldsymbol{\lambda}_{\xi}$, i.e., a "well-estimated" prior knowledge within BPA (in this case $\boldsymbol{\lambda}_{\xi}$), yields even more stable results than using a $\boldsymbol{\lambda}_{\mathcal{T}}$ estimated on much less data, even if such data is in-domain data (in this case $\boldsymbol{\lambda}_{\mathcal{D}}$). These results lead to the conclusion that using BPA instead of MERT or as a complementary post-process step is a good option in low-resource environments, even if this is not an adaptation problem any longer.

4.9 Conclusions and future work

In this chapter, Bayesian predictive adaptation has been thoroughly analysed for its application to statistical machine translation. On the one hand, the theoretical framework for adapting either the feature functions or the log-linear weights present in most state-of-the-art statistical machine translation systems has been developed. On the other hand, experimental

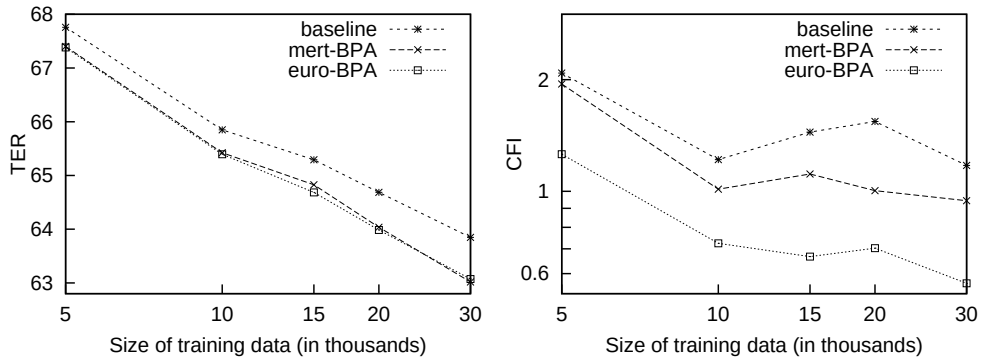


Figure 4.21: Translation quality and confidence interval sizes when using BPA as a stabilisation method. `baseline` computed by means of MERT, `mert-BPA` stands for the BPA approach when using the MERT-weights as mean vector within the BPA prior, and `euro-BPA` stands for the BPA approach when using the Europarl weights within the BPA prior. Both plots present the x-axis in log-scale and, in addition, the `CFI` plot also presents the y-axis in log-scale. The size of the training data is given in thousands of sentences.

results analysing the effectiveness of such adaptation procedures have been reported. In addition, three different scenarios have been studied where Bayesian adaptation can be applied: batch adaptation, online adaptation and system stabilisation.

Regarding the adaptation of the log-linear weights, results show that BPA has an interesting potential when the amount of adaptation data is relatively small. Consistent improvements in translation quality are obtained over the baseline system, as measured by TER, with as few as 10 adaptation samples, and up to an amount of adaptation data that allows a complete re-estimation of the model parameters. Results show that BPA, when applied to log-linear weight adaptation, proves to be more stable than MERT, which relies heavily on the amount of adaptation data and turns very unstable whenever few adaptation samples are available. It should be emphasised that an adaptation technique, by nature, is only useful whenever the amount of adaptation data is low, and our technique proves to behave well in such context. Whenever the amount of adaptation data is high, the best thing that one can do is to re-estimate the model parameters from scratch, although such re-estimation is often very costly. From a computational point of view, the Bayesian adaptation technique presented does not imply a significant computational overhead, the largest part of the computational complexity being taken by the sentence-level computation of the translation quality counts, which are required for the adaptation data. Hence, we consider that the technique presented here could easily be implemented within the decoder itself without a significant increase in computational complexity. We consider this important, since it implies that rerunning MERT for each adaptation set is not needed.

Different parameter sampling strategies have been studied when applying Bayesian predictive adaptation to the adaptation of the log-linear weights, such as Markov chain Monte Carlo, sampling from the Gaussian prior, an ad-hoc heuristic sampling strategy and the Viterbi

sampling approach. From the experimental results obtained, it emerges that the ad-hoc heuristic sampling strategy is able to perform at least as well as MCMC, and is computationally less expensive. Nevertheless, this heuristic strategy requires the introduction of an artificial meta-parameter δ because the probability distribution is not normalised. Such leveraging factor must be tuned beforehand. In contrast, the MCMC strategy does not require this δ, but appears to provide slightly less stable results.

Experimental results also show that BPA is an appropriate adaptation strategy for its application to the adaptation of log-linear weights in an on-line setup. In this context, interesting improvements in translation quality may be obtained without introducing a significant computational overhead (less than a second per sentence). Including such an adaptation capability is critical in environments where human translators work in collaboration with the SMT system, such as in an interactive machine translation scenario. A possible extension of the work presented here regards the assignment of a decaying weight to each sample within the sliding window (the adaptation sample) $\mathcal{A}_t$.

In addition, it has also been shown how to apply BPA in order to achieve more stability in the results achieved in conditions where bilingual data is very scarce. By adopting the best point-estimation of the model parameters as mean vector within the Gaussian prior, more stable results are achieved, while yielding improvements in translation quality as well. Adopting as mean vector an external, canonical set of parameters which may be assumed to be well estimated provides even more stability to the results.

Regarding the adaptation of the feature functions, experiments conducted in this direction are not very encouraging: although not negative, the computational overhead introduced is not justified by the very limited improvements in translation quality achieved. One possible reason for this may be that current state-of-the-art SMT systems act more like a memory-based MT system, rather than a fully-fledged statistical system with properly estimated statistical distributions. As pointed out in Section 2.7.2, if the final amount of phrase pairs that actually have a competing phrase (i.e., the number of phrases that are not chosen deterministically) is very low, re-estimating $\boldsymbol{h}$ is bound to have a very small effect, if any. Another possible reason might be that there are too many parameters to be adapted, in which case several strategies could be followed in order to solve both the sparsity problems derived and the problem involving the high computational overhead. In the first place, it would be interesting to research possible ways of binding the parameters present in the phrase-table, such as using unsupervised clustering algorithms or grouping the different bilingual phrases according to their part-of-speech tags. Another possible strategy for confronting this problem is to make use of the different phrase-table reduction techniques that are present in the literature, such as the ones described in Chapter 2 or the ones presented in (Eck et al., 2007; Johnson et al., 2007).

The author would like to thank Dr. Nicola Cancedda for his very helpful comments on a previous version of this chapter, which led to improving the contents in a very significant manner.

The work presented in this chapter was accepted for publication in an international conference and an international workshop, respectively:

- **G. Sanchis-Trilles** and F. Casacuberta. Bayesian Adaptation for Statistical Machine Translation. In *Proceedings of the Joint IAPR International Workshops on Structural*

and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition, S+SSPR 2010, pages 620–629, Çesme, Izmir (Turkey), August 2010.

- **G. Sanchis-Trilles** and F. Casacuberta. Log-linear weight optimisation via Bayesian Adaptation in Statistical Machine Translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (poster volume), COLING 2010*, pages 1077-1085, Beijing (China), August 2010.

In addition, the stabilisation strategy described in Section 4.5 was used within the system presented for an international MT competition:

- G. Gascó, V. Alabau, J. Andrés–Ferrer, J. González-Rubio, M. A. Rocha, **G. Sanchis-Trilles**, F. Casacuberta, J. González and J. A. Sánchez. ITI-UPV system description for IWSLT 2010 In *Proceedings of the 2010 International Workshop on Spoken Language Translation, IWSLT 2010*, pages 85–92, Paris (France), December 2010.

Furthermore, the online variant of BPA presented, together with other work on online adaptation, has also been accepted for publication in an international journal:

- P. Martínez-Gómez, **G. Sanchis-Trilles** and F. Casacuberta. Online adaptation strategies for statistical machine translation in post-editing scenarios. In *Pattern Recognition*. (*In press*)

Lastly, most of the work presented in this chapter has been submitted to an international journal:

- **G. Sanchis-Trilles** and F. Casacuberta. Batch and online Bayesian predictive adaptation in statistical machine translation. (submitted for revision)

Bibliography

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 355–362, July 27–31 2011.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, and Enrique Vidal. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28, 2009.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Jonathan H. Clark, Chris Dyer, and Noah A. Smith. Better hypothesis testing for machine translation: Controlling for optimizer instability. In *Proceedings of the annual conference of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, June 19–24 2011.

Hal Daume III and Jagadeesh Jagarlamudi. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the annual conference of the Association for Computational Linguistics: Human Language Technologies*, pages 407–412, June 19–24 2011.

John DeNero, Alexandre Bouchard-Côté, and Dan Klein. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 314–323, October 25–27 2008.

Richard Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

Kevin Duh, Katsuhito Sudoh, Tomoharu Iwata, and Hajime Tsukada. Alignment inference and bayesian adaptation for machine translation. In *Proceedings of the Machine Translation Summit X*, pages 19–23, September 19–23 2011.

Matthias Eck, Stephan Vogel, and Alex Waibel. Translation model pruning via usage statistics for statistical machine translation. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 21–24, April 22–27 2007.

George Foster, Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 451–459, October 9–11 2010.

Guillem Gascó, Vicent Alabau, Jesús Andrés-Ferrer, Jesús González-Rubio, Martha-Alicia Rocha, Germán Sanchis-Trilles, Francisco Casacuberta, Jorge González, and Joan-Andreu Sánchez. Iti-upv system description for iwslt 2010. In *Proceedings of the international Workhop on Spoken Language Translation*, pages 85–92, December 2–3 2010.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

W. Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):19–109, 1970.

Qiang Huo, Chorkin Chan, and Chin-Hui Lee. Bayesian adaptive learning of the

parameters of hidden markov model for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 3(5):334–345, 1995.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. Improving translation quality by discarding most of the phrasetable. In *Proceedings the conference on Empirical Methods in Natural Language Processing and Computational Natural Language Processing*, pages 967–975, June 28–30 2007.

Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel. Bayesian adaptation revisited. In *Proceedings of ISCA ITRW ASR2000: Automatic Speech Recognition: Challenges for the new Millennium*, pages 112–119, September 18–20 2000.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the international conference on Acoustics, Speech and Signal Processing*, pages 181–184, May 9–12 1995.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2010. ISBN 978-0521874151.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics: Demo and Poster Sessions*, pages 177–180, June 23–30 2007.

Pascual Martínez-Gómez, Germán Sanchis-Trilles, and Francisco Casacuberta. Passive-aggressive for on-line learning in statistical machine translation. In *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, pages 240–247, June 8–10 2011.

Coskun Mermer and Murat Saraclar. Bayesian word alignment for statistical machine translation. In *Proceedings of the annual conference of the Association for Computational Linguistics: Human Language Technologies*, pages 182–187, June 19–24 2011.

Franz J. Och. Minimum error rate training for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 160–167, July 7–12 2003.

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

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. Online learning for interactive statistical machine translation. In *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 546–554, June 2–4 2010.

Michael Paul, Marcello Federico, and Sebastian Stüker. Overview of the IWSLT 2010 evaluation campaign. In *Proceedings of the international Workhop on Spoken Language Translation*, pages 3–27, December 2–3 2010.

Sujith Ravi and Kevin Knight. Bayesian inference for zodiac and other homophonic ciphers. In *Proceedings of the annual conference of the Association for Computational Linguistics: Human Language Technologies*, pages 239–247, June 19–24 2011.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, September 16–20 2002.

Charles Sutton and Andrew McCallum. *An introduction to conditional random fields for relational learning*. MIT Press, 2006.

Fabio Valente and Christian J. Wellekens. Variational bayesian adaptation for speaker clustering. In *Proceedings of the international conference on Acoustics, Speech and Signal Processing*, pages 965–968, March 18–23 2005.

Kai Yu and Mark J. Gales. Bayesian adaptation and adaptively trained systems. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 209–214, November 27 – December 1 2005.

Kai Yu and Mark J. Gales. Incremental adaptation using bayesian inference. In *Proceedings of the international conference on Acoustics, Speech and Signal Processing*, pages 217–220, May 15–19 2006.

Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of the annual conference of the Association for Computational Linguistics: Human Language Technologies*, pages 97–105, June 15–20 2008.

CHAPTER 5

Enriching user-machine interaction in IMT

Probleme kann man niemals mit derselben Denkweise lösen, durch die sie entstanden sind.
Albert Einstein

Contents

5.1 Introduction 129
5.2 Related work 129
5.3 Anticipated proposal as a form of weaker feedback 130
5.4 Partial refusal pointer action 131
5.5 Experimental results 133
5.6 Conclusions and future work 135
Bibliography 137

EINSTEIN: »Man sperrt uns ein wie wilde Tiere!«
MÖBIUS: »Wir sind wilde Tiere. Man darf uns nicht auf die Menschheit loslassen.«
NEWTON: »Gibt es wirklich keinen andern Ausweg?«
MÖBIUS: »Keinen.«
EINSTEIN: »Johann Wilhelm Möbius. Ich bin ein anständiger Mensch. Ich bleibe.«
NEWTON: »Ich bleibe auch. Für immer.«
MÖBIUS: »Ich danke euch. Um der kleinen Chance willen, die nun die Welt doch noch besitzt davonzukommen.« *Er erhebt sein Glas.* »Auf unsere Krankenschwestern!«
Sie haben sich feierlich erhoben.
[...]
Sie trinken, stellen die Gläser auf den Tisch.
NEWTON: »Verwandeln wir uns wieder in Verrückte. Geistern wir als Newton daher.«
EINSTEIN: »Fiedeln wie wieder Kreisler und Beethoven.«
MÖBIUS: »Lassen wir wieder Salomo erscheinen.«
NEWTON: »Verrückt, aber weise.«
EINSTEIN: »Gefangen, aber frei.«
MÖBIUS: »Physiker, aber unschuldig.«

Die Physiker. Friedrich Dürrenmatt.

EINSTEIN: "They locked us like wild animals!"
MOBIUS: "We are wild animals. We must not let ourselves to humanity."
NEWTON: "Is there really no other way"?
MOBIUS: "No".
EINSTEIN: "Johann Wilhelm Möbius. I am a decent person. I'm staying."
NEWTON: "I will stay. For good."
*MOBIUS: "I thank you. To the small chance of sake, which is now the world has yet get away. "*He raised his glass. *"To our nurses!"*
You have risen solemnly.
[...]
They drink, the glasses on the table.
NEWTON: "Turn us back into lunatics. Therefore, we Spirits as Newton."
EINSTEIN: "fiddles again as Kreisler and Beethoven."
MOBIUS: "Let reappear Solomon."
NEWTON: "Crazy, but wise."
EINSTEIN: "Trapped, but free."
MOBIUS: "Physicist, but innocent. "

The physicist. Google Translate.

5.1 Introduction

Interactive Machine Translation was first introduced within the TransType (Foster et al., 1997; Langlais et al., 2000, 2002) project, where it proved to be able to deliver interesting benefits to potential users, by considerably reducing the effort needed in order to translate a complete text. Nevertheless, one aspect which has remained mostly unchanged since those first approaches to IMT is the user–machine interaction protocol: traditional IMT systems only received feedback whenever the user typed in a new word. However, such protocol accepts many improvements. In the present chapter, we show how to enrich user–machine interaction by making use of weaker feedback. Specifically, two types of *pointer actions* (PAs) are considered here as weaker feedback. The first one, which we have named *anticipated proposal* , proposes to observe the actions that the user performs before modifying a given hypothesis, with the purpose of anticipating such modification. The second kind of weaker feedback consists in allowing the user to simply state that he does not like the (partial) hypothesis provided, and that he wants it to be replaced. This latter kind of feedback will be referred to as partial refusal. Both of these interaction capabilities will be implemented in the present chapter by means of a pointer action (PA), although one could easily picture other devices for performing these kind of actions.

The rest of this chapter is structured as follows. Section 5.2 briefly reviews similar work. Then, Section 5.3 details the main idea behind considering pointer actions as an additional information source for the system, and how it is possible to take advantage of the actions the user is performing even when no keyboard action is performed. Next, in Section 5.4, an additional twist to pointer actions is detailed so as to offer the user different explicit interaction possibilities, with the purpose of reducing the number of times the user will need to introduce additional words. Experimental results are presented in Section 5.5, in which an IMT environment is simulated with the purpose of assessing the benefits that can be achieved by means of the two different PAs presented. Finally, the conclusions that can be drawn from the work presented in this chapter are detailed in Section 5.6, together with possible extensions that will be conducted as future work.

5.2 Related work

A work that is very similar to the one described here was performed in (Romero et al., 2009). However, such work researched the use of weaker feedback within an interactive handwritten text recognition scenario, and not in an IMT setting as is the case in the present chapter. In addition, weaker feedback has been also researched for interactive text generation (Ruiz, 2010), where the main goal is to help handicapped people to communicate in cases where they might have lost the ability to do so by other means such as writing, oral communication or typing.

Even though the work presented here does not take advantage of a multimodal setting, other works exist, in which the classical IMT framework is expanded by taking advantage of multimodality. For instance, (Alabau et al., 2011) propose the use of a speech recognition system with the purpose of allowing the human user to correct the errors made by the IMT system by simply stating, orally, where such errors were made. However, instead of allowing

SOURCE $(\boldsymbol{x})$**:**		Para encender la impresora:
REFERENCE $(\boldsymbol{y})$**:**		To power on the printer:
ITER-0	$(\boldsymbol{p})$	( )
	$(\hat{\boldsymbol{s}}_h)$	*To switch on:*
ITER-1	$(\boldsymbol{p})$	To
	$(\boldsymbol{s}_l)$	|*switch on:*
	$(\hat{\mathbf{s}}_h)$	*power on the printer:*
ITER-2	$(\boldsymbol{p})$	To power on the printer:
	$(\boldsymbol{s}_l)$	( )
	(k)	(#)
	$(\hat{\boldsymbol{p}}_h)$	( )
FINAL	$(\boldsymbol{p} \equiv \boldsymbol{y})$	To power on the printer:

Figure 5.1: Example of anticipated proposal pointer action which solves an error of a missing word. In this case, the system produces the correct suffix $\boldsymbol{s}_h$ immediately after the user validates a prefix $\boldsymbol{p}$, implicitly indicating that we wants the suffix to be changed, without need of any further action. In **ITER-1**, character | indicates the position where a pointer action was performed, $\boldsymbol{s}_l$ is the suffix which was rejected by that pointer action, and $\hat{\boldsymbol{s}}_h$ is the new suffix that the system suggests after observing that $\boldsymbol{s}_l$ is to be considered incorrect. Character # is a special character introduced by the user to indicate that the hypothesis is to be accepted.

the speech recognition full freedom when recognising the corrections of the user, such recognition was biased by the translation model in such a way, that the best scoring suffixes are those that are most probable according to both the SMT system and the speech recognition system. In related work, (Alabau et al., 2011) propose a similar scenario, but allowing the user to correct the errors by means of a graphic tablet or screen, with which the user may interact by writing in some word, or even just some kind of gesture.

5.3 Anticipated proposal as a form of weaker feedback

The key idea behind considering pointer actions as an additional communication vehicle between the system and the user is that, in order to correct a hypothesis, the user first needs to position the cursor in the place where he wants to type a word, be it for correcting it, for introducing a new word, or for deleting an existing one. In this case, we will assume that this is done by performing a pointer action. By doing so, the user is already providing a very valuable information to the system. Namely, he is signalling that whatever information is located before the cursor is to be considered as correct, hence validating the current prefix $\boldsymbol{p}$. More importantly, however, he is also signalling that he does not like whatever word comes after $\boldsymbol{p}$, and that he is about to change it. At this point, the system can capture this fact and, knowing that such suffix is to be considered as incorrect, provide a new translation hypothesis in which the prefix remains unchanged and the suffix is replaced by a new one in which the first word is different to the first word of the previous suffix.

An example of such behaviour can be seen in Figure 5.1. In this example, the SMT system

first provides a translation which the user does not like. Hence, he positions the cursor before word "*switch*", with the purpose of typing in "*power*". By doing so, he is validating the prefix "*To*", and signalling that he wants "*switch*" to be replaced. Before typing in anything, the system realises that he is going to change the word located after the cursor, and replaces the suffix by another one, which is the one the user had in mind in the first place. Finally, the user only has to accept the final translation.

Obviously, having the system change the incorrect suffix does not mean that the new suffix will be correct. However, given that the system knows that the first word in the current suffix is incorrect, the worst case would only imply that the newly introduced word would still be incorrect. This entails that the user would need to type in the correct word, as he was going to do anyway. However, if the new proposed suffix happens to be correct, the system will have spared the user one interaction, which is typing in the new word, and the user will happily find that he only needs to accept such word, or perhaps even the complete suffix.

We are naming this kind of pointer action anticipated proposal because the user does not need to perform an explicit action in order to inform the system that it needs to change the suffix: it is the system itself who realises that the user is going to type in a word and anticipates the user's intentions, suggesting a new suffix hypothesis. For this reason, and given the fact that the user would need to position the cursor anyway, it is important to point out that any improvement achieved by this kind of pointer action is an improvement *per se*, since it requires no further effort from the user. For this reason, it is assumed to have no cost.

The anticipated proposal pointer action can be formulated as: Given a source sentence $\boldsymbol{x}$, a consolidated prefix $\boldsymbol{p}$ and a suffix $\boldsymbol{s}'$ suggested by the system in the previous interaction, search for another suffix $\hat{\boldsymbol{s}}$ such that the first word in $\hat{\boldsymbol{s}}$ is different from the first word in $\boldsymbol{s}'$

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}:\boldsymbol{s}_1 \neq \boldsymbol{s'}_1}{\operatorname{argmax}} P(\boldsymbol{s}|\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{s}') \tag{5.1}$$

5.4 Partial refusal pointer action

In contrast to anticipated proposal pointer actions, one could easily picture a scenario where the user simply wants a given suffix to be changed, without taking into consideration whether the cursor is already located just in front of the first erroneous word. Assuming that the underlying IMT system is efficient enough when attempting to provide high quality suffixes, the human expert would just need to click before the first word of the suffix he intends to change in order to have it replaced without any further action. This pointer action is named partial refusal because the user needs to explicitly ask the system for another hypothesis by means of a pointer action, whereas in the case of anticipated proposal pointer actions the user only performed a pointer action whenever he needed to position the cursor before typing. Obviously, this could also be done by using some other different device, but in this case we assume this is done using the mouse. Note that this kind of pointer action does imply an added cost, since the user needs to perform an explicit action for signalling the system that he wants the suffix to be replaced. However, if the underlying MT engine providing suffixes is powerful enough, the benefit obtained may easily be worth the hassle, since performing a pointer action is less costly than introducing one (or several) whole new word. Of course, in this kind of pointer action the system is expecting a participative and collaborative attitude

SOURCE $(\boldsymbol{x})$**:** Seleccione el tipo de instalación.
REFERENCE $(\boldsymbol{y})$**:** Select the type of installation.

ITER-0	$(\boldsymbol{p})$	( )
	$(\hat{\boldsymbol{s}}_h)$	*Select the installation wizard.*
ITER-1	$(\boldsymbol{p})$	Select the
	$(\boldsymbol{s}_l)$	|*installation wizard.*
	$(\hat{\boldsymbol{s}}_h)$	*install script.*
ITER-2	$(\boldsymbol{p})$	Select the
	(k)	type
	$(\hat{\boldsymbol{s}}_h)$	*installation wizard.*
ITER-3	$(\boldsymbol{p})$	Select the type
	$(\boldsymbol{s}_l)$	|*installation wizard.*
	$(\hat{\boldsymbol{s}}_h)$	*of installation.*
ITER-4	$(\boldsymbol{p})$	Select the type of installation.
	$(\boldsymbol{s}_l)$	( )
	(k)	(#)
	$(\hat{\boldsymbol{s}}_h)$	( )
FINAL	$(\boldsymbol{p} \equiv \boldsymbol{y})$	Select the type of installation.

Figure 5.2: Example of partial refusal pointer action which corrects an erroneous suffix. In this case, an anticipated proposal pointer action is performed in **ITER-1** with no success. Hence, the user introduces word "*type*" in **ITER-2**, which leaves the cursor position located immediately after word "*type*". In this situation the user would not need to perform a pointer action to re-position the cursor and continue typing in order to further correct the remaining errors, since he could simply continue typing the word he has in mind. However, since he has learnt the potential benefit of pointer actions, he performs a partial refusal pointer action in order to ask for a new suffix hypothesis, which happens to correct the error.

from the user, which was not the case in the case of anticipated proposal weaker feedback. An example of such an explicit pointer action correcting an error can be seen in Figure 5.2

In this case, however, there is a cost associated to this kind of pointer actions, since the user does need to perform additional actions, which may or may not be beneficial. It is very possible that, even after asking for several new hypothesis, the user will even though need to introduce the word he had in mind, hence wasting the additional pointer actions he had performed.

Assuming the user has already performed n pointer actions until the current moment and is demanding yet another suffix $\hat{\boldsymbol{s}}$ from the system, the partial refusal problem can be formalised in a very similar way to the case of anticipated proposal pointer actions:

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s}:\boldsymbol{s}_1 \neq \boldsymbol{s}_1^{(i)}, \forall i \in \{1..n\}}{\operatorname{argmax}} P(\boldsymbol{s}|\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(n)}) \tag{5.2}$$

where $\boldsymbol{s}_1^{(i)}$ is the first word of the i-th suffix discarded, and $\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(n)}$ is the set of all n suffixes discarded.

Note that this kind of pointer action could also be implemented with some other kind of

interface, e.g. by typing some special key such as `F1` or `Tab`. However, the experimental results would not differ, and in the existing user interface it seemed more intuitive to implement it as a pointer action.

5.5 Experimental results

In addition to WSR, and because pointer actions are also introduced as a new action, results in terms of *Pointer Action Ratio* (PAR) will also be reported. PAR is the quotient between the amount of partial refusal pointer actions performed and the number of words of the final translation. Hence, the purpose is to elicit the number of times the user needed to request a new translation (i.e. performed a pointer action), on a per word basis.

Also for the case of partial refusal pointer actions, results in terms of uPAR (useful PAR) will also be reported. uPAR indicates the amount of pointer actions which were *useful*, i.e. the pointer actions that actually produced a change in the first word of the suffix and such word was accepted. Formally, uPAR is defined as follows:

$$uPAR = \frac{PAC - n \cdot WSC}{PAC} \tag{5.3}$$

where PAC stands for "Pointer Action Count" (the total number of pointer actions performed), WSC for "Word Stroke Count" (the total number of word strokes performed) and n is the maximum amount of pointer actions allowed before the user types in a word. Note that $PAC - n \cdot WSC$ is the amount of pointer actions that were useful since WSC is the amount of word-strokes the user performed even though he had already performed n pointer actions, i.e., $n \cdot WSC$ is the number of *useless* PAs.

Since WSR and PAR will be used with a single reference, the results presented here are clearly pessimistic. In fact, it is relatively common to have the underlying SMT system provide a perfectly correct translation, which is "corrected" by the IMT procedure into another equivalent translation, increasing WSR and PAR significantly by doing so.

Experiments were conducted on the Europarl corpus, in the partition established for the WMT08 (see Section 1.4). Specifically, the language pairs studied were Spanish $\rightarrow$ English, French $\rightarrow$ English and German $\rightarrow$ English.

As a first step, an SMT system was trained for each of the language pairs cited in the previous subsection. This was done by means of the Moses toolkit (Koehn et al., 2007), and the weights $\boldsymbol{\lambda}$ of the log-linear model were optimised by means of MERT.

This being done, word graphs were generated for the IMT system. For this purpose, the multi-stack phrase-based decoder which is part of the Thot toolkit (Ortiz-Martínez et al., 2005) (see Section 1.5) was employed. The Moses decoder was discarded in this case because preliminary experiments performed with it revealed that the decoder by (Ortiz-Martínez et al., 2005) performs clearly better when generating word graphs for their use in IMT. In addition, an experimental comparison in regular SMT with the Europarl corpus found that the performance difference between both decoders was negligible. However, it must be noted that the experiments performed in this chapter were carried out in year 2008, and since then the quality of the word graphs provided by the Moses decoder has greatly improved (the version used at that time was checked out from the official subversion repository on November

Table 5.1: WSR improvement when considering non-explicit MAs. "rel." indicates the relative improvement. All results are given in %.

pair	baseline	ant. proposal	rel.
Es–En	63.0±0.9	59.2±0.9	6.0±1.4
En–Es	63.8±0.9	60.5±1.0	5.2±1.6
De–En	71.6±0.8	69.0±0.9	3.6±1.3
En–De	75.9±0.8	73.5±0.9	3.2±1.2
Fr–En	62.9±0.9	59.2±1.0	5.9±1.6
En–Fr	63.4±0.9	60.0±0.9	5.4±1.4

13, 2007). The decoder was set to only consider monotonic translation, since in real IMT scenarios considering non-monotonic translation leads to excessive waiting time for the user.

Finally, the word graphs obtained were used within the IMT procedure to produce the reference translation contained in the test set, measuring WSR and PAR. The results of such a setup can be seen in Table 5.1. As a baseline system, the traditional IMT framework presented in Section 1.3 is reported, in which no pointer action is taken into account. Then, anticipated proposal pointer actions were introduced, obtaining an average improvement in WSR of about 3.2% (4.9% relative). The table also shows the confidence intervals at a confidence level of 95%. These intervals were computed following the bootstrap technique described in Section 1.2.2. Since the confidence intervals do not overlap, it can be stated that the improvements obtained are statistically significant.

Once the anticipated proposal pointer actions were considered and introduced into the system, the effect of performing up to a maximum of 5 partial refusal pointer actions was analysed, taking as baseline system this time the one that already includes anticipated proposal pointer actions. Here, the user was modelled in such a way that, in case a given word is considered incorrect, he will always ask for another translation hypothesis until he has asked for as many different suffixes as pointer actions considered. The results of this setup can be seen in Figure 5.3. This yielded a further average improvement in WSR of about 16% (25% relative improvement) when considering a maximum of 5 explicit pointer actions. However, relative improvement in WSR and uPAR drops significantly when increasing the maximum allowed amount of explicit pointer actions from 1 to 5. For this reason, it is difficult to imagine that a user would perform more than two or three pointer actions before actually typing in a new word. Nevertheless, just by asking twice for a new suffix before typing in the word he has in mind, the user might be saving about 15% of word-strokes.

Although the results in Figure 5.3 are only for the translation direction "foreign"→English, the experiments in the opposite direction (i.e. English→"foreign") were also performed. However, the results were very similar to the ones displayed here. Because of this, and for clarity purposes, we decided to omit them and only display the direction "foreign"→English.

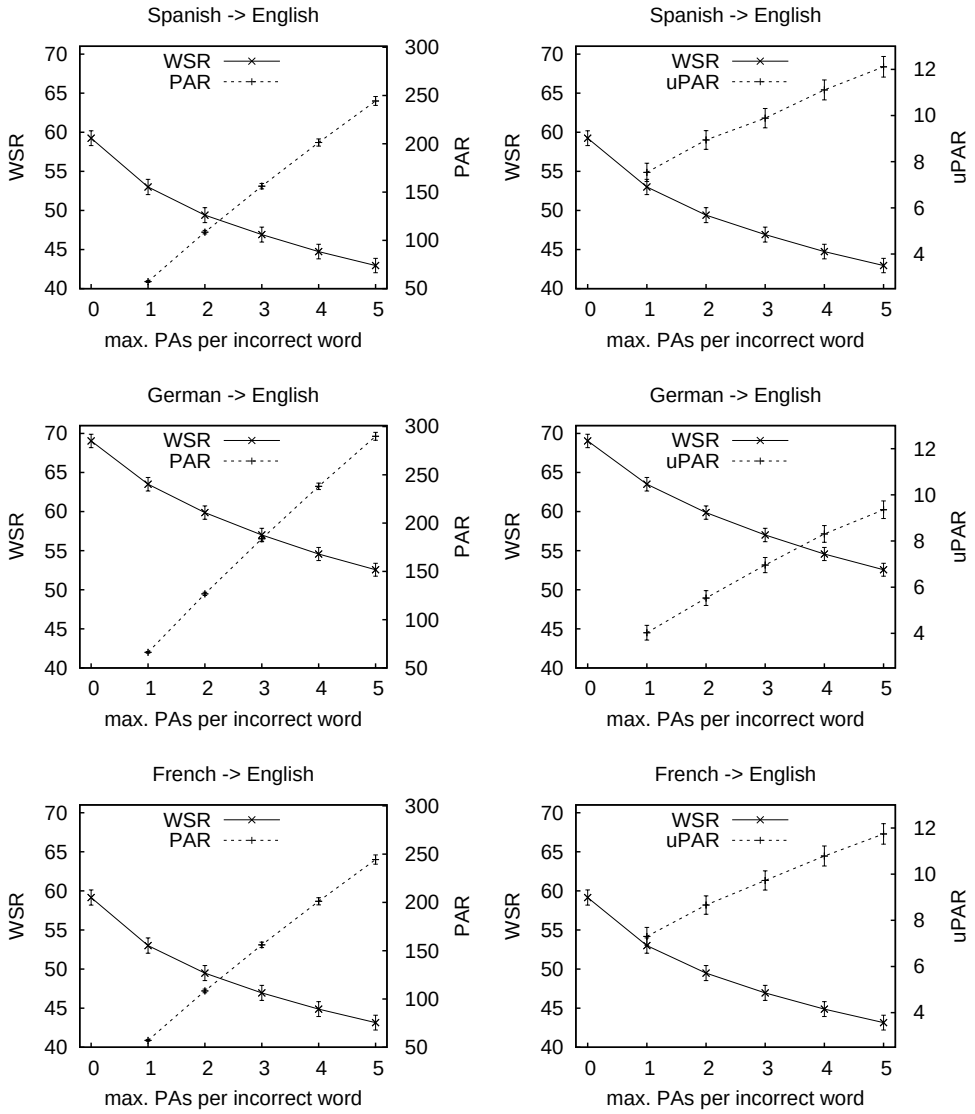


Figure 5.3: WSR improvement when considering one to five maximum PAs. All figures are given in %. The left column lists WSR improvement versus PAR degradation, and the right column lists WSR improvement versus uPAR. Confidence intervals at 95% confidence level following (Koehn, 2004).

5.6 Conclusions and future work

In this chapter, new input sources for IMT have been introduced. By considering pointer actions as a form of weaker feedback, it has been shown that a significant benefit can be

obtained, in terms of word-stroke reduction, both when considering only anticipated proposal pointer actions and when considering pointer actions as a way of offering the user several suffix hypotheses (i.e.,partial refusal). In addition, these ideas have been applied on a state-of-the-art SMT baseline, such as phrase-based models. To achieve this, word graphs were first obtained for each sentence which is to be translated. Experiments were carried out on a reference corpus in SMT.

Note that there are other systems (Esteban et al., 2004) that, for a given prefix, provide n-best lists of suffixes. Although it might seem that such approach is very similar to the one presented here, the functionality of the present system is slightly (but fundamentally) different, since the suggestions are demanded to be different in their first word, which implies that the n-best list is scanned deeper, going directly to those hypotheses that may be of interest to the user. In addition, this can be done "on demand", which implies that the system's response is faster and that the user is not confronted with a large list of hypotheses, which often results overwhelming.

As future work, a human evaluation would be necessary to assess the appropriateness of the improvements described.

The work presented in this chapter was accepted for publication in an international conference:

- **G. Sanchis-Trilles**, Daniel Ortiz-Martínez, Jorge Civera, Francisco Casacuberta, Enrique Vidal and Hieu Hoang Improving Interactive Machine Translation via Mouse Actions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008*, pages 485–494, Honolulu, Hawaii (USA), October 2008.

In addition, it also lead to a publication in an international workshop:

- **G. Sanchis-Trilles**, M.T. González, F. Casacuberta, E. Vidal and J. Civera Introducing Additional Input Information into IMT Systems. In *Proceedings of the 5th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, MLMI 2008*, pages 284–295, Utrecht (The Netherlands), September 2008.

Furthermore, currently there is work in progress for publishing an article in an international journal, together with similar work done by another author in the field of interactive text recognition.

Bibliography

Vicent Alabau, Luis Rodríguez-Ruiz, Alberto Sanchis, Pascual Martínez-Gómez, and Francisco Casacuberta. On multimodal interactive machine translation using speech recognition. In *Proceedings of the International Conference on Multimodal Interaction*, pages 129–136, November 14–18 2011.

José Esteban, José Lorenzo, Antonio S. Valderrábanos, and Guy Lapalme. Transtype2 - an innovative computer-assisted translation system. In *Proceedings of the annual meeting of the Association for Computational Linguistics*, pages 94–97, July 21–26 2004.

George Foster, Pierre Isabelle, and Pierre Plamondon. Target-text mediated interactive machine translation. *Machine Translation*, 12(1–2):175–194, 1997.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 388–395, July 25–26 2004.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics: Demo and Poster Sessions*, pages 177–180, June 23–30 2007.

Philippe Langlais, George Foster, and Guy Lapalme. Unit completion for a computer-aided translation typing system. *Machine Translation*, 15(4):267–294, 2000.

Philippe Langlais, Guy Lapalme, and Marie Loranger. Transtype: Development-evaluation cycles to boost translator's productivity. *Machine Translation*, 15(4):77–98, 2002.

Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. Thot: a toolkit to train phrase-based statistical translation models. In *Proceedings of the Machine Translation Summit X*, pages 141–148, September 12–16 2005.

Verónica Romero, Alejandro H. Toselli, and Enrique Vidal. Using mouse feedback in computer assisted transcription of handwritten text images. In *Proceedings of the international Conference on Document Analysis and Recognition*, pages 96–100, July 26–29 2009.

Luis Rodríguez Ruiz. *Interactive Pattern Recognition applied to Natural Language Processing*. PhD thesis, Universitat Politècnica de València, 2010.

CHAPTER 6

Conclusions

La inspiración existe, pero tiene que encontrarte trabajando.

Pablo Ruiz Picasso

Contents

6.1 Summary **141**
6.2 Scientific publications **142**
6.3 Future work **145**
Bibliography **147**

Asnografía

Leo en un Diccionario: Asnografía, *s.f.: Se dice, irónicamente, por descripción del asno*.

¡Pobre asno! ¡Tan bueno, tan noble, tan agudo como eres! Irónicamente... ¿Por qué? ¿Ni una descripción seria mereces, tú, cuya descripción cierta sería un cuento de primavera? ¡Si al hombre que es bueno debieran decirle asno! ¡Si al asno que es malo debieran decirle hombre! Irónicamente... De ti, tan intelectual, amigo del viejo y del niño, del arroyo y de la mariposa, del sol y del perro, de la flor y de la luna, paciente y reflexivo, melancólico y amable, Marco Aurelio de los prados...

Platero, que sin duda comprende, me mira fíjamente con sus ojazos lucientes, de una blanda dureza, en los que el sol brilla, pequeñito y chispeante, en un breve y convexo firmamento verdinegro. ¡Ay! ¡Si su peluda cabezota idílica supiera que yo le hago justicia, que yo soy mejor que esos hombres que escriben Diccionarios, casi tan bueno como él!

Y he puesto al margen del libro: Asnografía, *sentido figurado: Se debe decir, con ironía ¡claro está!, por descripción del hombre imbécil que escribe Diccionarios.*

Platero y Yo. Juan Ramón Jiménez.

I read in a Dictionary: Asnografía, nd: It is said, ironically, by description of the donkey.

Poor donkey! So good, so noble, so sharp you are! Ironically... Why? Not even a description would deserve it, you, whose story would be a true description of spring? If the man who should say good ass! If it's bad ass man should say! Ironically... From you, so intellectual, friend of the old and the child, the stream and the butterfly, sun, and the dog, flower and moon, patient and thoughtful, melancholy and gentle, Marco Aurelio of meadows...

Platero, which undoubtedly includes, stares at me with her big eyes shining, a soft hardness, where the sun shines, tiny, sparkling in a short and convex green-black sky. Oh! If your furry idyllic stubborn I do know that justice, that I am better than the men who write dictionaries, almost as good as him!

And I put the book aside: Asnografía, figurative sense: It must be said, with irony of course!, For description of the man who writes dictionaries idiot.

Platero y Yo. Google Translate.

6.1 Summary

The work developed in this thesis confronts three of the main problems present in state-of-the-art statistical machine translation systems, and which prevent their wide-spread use within current computer assisted translation tools. These topics are efficiency, adaptability, and usability.

Striving for decreasing the response time of state-of-the-art machine translation systems, we presented a novel technique for pruning the total amount of parameters present in the translation system. The intuition behind this technique is to obtain one single segmentation of each bilingual sentence present in the training data, arriving to a full re-estimation of the model parameters. With the purpose of reducing the possible effects on translation quality, n-best segmentations are also considered. In statistical machine translation, experimental results show that a very aggressive pruning may be performed without any loss at all in translation quality, achieving very important speedup rates. The experiments were carried out by using state-of-the-art statistical machine translation systems, covering several different language pairs, and with corpora used in standard machine translation tasks. In interactive machine translation, the performance gains achieved without any loss in system performance are less impressing, although not negligible at all. However, it must be kept in mind that the experiments performed in this direction were carried out in a simulated interaction setting, with only one reference translation, which implies that the evaluation metric used has a very important impact on the results obtained.

When confronting the adaptability problem, two different research directions were explored. On the one hand, we developed a strategy for increasing the adaptability of the language model, which is a key component of every machine translation system. This technique is inspired by the idea of increasing the flexibility of the language model by subdividing it into several, more specific, sub-models. Such models were constructed either by taking advantage of supervised labels concerning dialogue act information, when such labels are available, or by building unsupervised clusters of the available training data. The results obtained on different standard machine translation tasks point towards a potential benefit which can be achieved by applying the technique described. Even though the improvements obtained in the work presented here are relatively limited, these are coherent throughout all the experiments performed, involving different corpora and language pairs.

On the other hand, adaptability was also pursued by dealing with the translation model adaptation problem from the Bayesian perspective. In this context, Bayesian predictive adaptation is unveiled as a powerful adaptation method in statistical machine translation, with a statistically sound formulation, allowing an efficient implementation, and which entails consistent and coherent improvements. Experiments performed on standard corpora in statistical machine translation and with state-of-the-art systems have proved that the adaptation of the log-linear weights present in modern models is an effective way of adapting the translation model, yielding important improvements in translation quality even when the amount of adaptation data is very low. However, adapting the feature functions led to a less promising result, since the additional computational burden does not justify the marginal (yet coherent) improvements obtained.

Finally, concerning the usability of modern interactive machine translation systems, we have presented a simple, yet effective, extension of the traditional interaction scheme. The

key idea that has led to this extension involves realising that the the human translator is not only interacting with the translation system by means of the keyboard. In this sense, we have presented the mouse as a valuable information supplier, both in an implicit and in an explicit way. On the one hand, we have shown how to anticipate the possible changes the user might want to perform, and on the other hand we have shown how to enrich the information facilitated to the user, while preventing clogging the interface with too much information. Experimental results in a simulated interactive machine translation scenario show that there is much to be gained by adopting the ideas described in this direction.

To summarise, the main contributions of this thesis are the following:

1. It is shown that the phrase-table present in state-of-the-art statistical machine translation systems can be aggressively pruned without any loss in translation quality. We present a technique for doing this, which evolves to a parameter re-estimation method.

2. Language model mixtures are presented as a promising way of providing flexibility to the language model. Results reported on different tasks point toward potential benefits.

3. Bayesian predictive adaptation is applied to statistical machine translation. The theoretical framework for achieving this is presented, and experimental results on different corpora prove that substantial improvements can be achieved. More specifically, the adaptation of the log-linear weights provides consistent improvements, while adapting the feature functions provides only marginal improvements.

4. The traditional interactive machine translation interface is improved by taking into account the mouse with which the user is able to perform different actions. By doing so, it is possible to improve the productivity achieved by a human translator in about 15%.

6.2 Scientific publications

Even though the scientific publications derived from this thesis have already been listed in their corresponding chapters, at this point we would like to summarise them, but listed according to their importance, rather than their research area.

First, an article was published in an international journal, with an estimated impact factor in year 2010 of 2.607:

- P. Martínez-Gómez, **G. Sanchis-Trilles** and F. Casacuberta. Online adaptation strategies for statistical machine translation in post-editing scenarios. In *Pattern Recognition*. (*In press*) (*Relative to Chapter 4*)

In addition, several research articles have been published in international conferences ranked A by the Computing Research and Education Association of Australasia (CORE):

- **G. Sanchis-Trilles**, Daniel Ortiz-Martínez, Jorge Civera, Francisco Casacuberta, Enrique Vidal and Hieu Hoang Improving Interactive Machine Translation via Mouse Actions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008*, pages 485–494, Honolulu, Hawaii (USA), October 2008. (*Relative to Chapter 5*)

- **G. Sanchis-Trilles** and F. Casacuberta. Bayesian Adaptation for Statistical Machine Translation. In *Proceedings of the Joint IAPR International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition, S+SSPR 2010*, pages 620–629, Çesme, Izmir (Turkey), August 2010. (*Relative to Chapter 4*)

- J. Andrés-Ferrer, **G. Sanchis-Trilles** and F. Casacuberta Similarity Word-Sequence Kernels for Sentence Clustering. In *Proceedings of the 8th International Workshop on Statistical Pattern Recognition, S+SSPR 2010*, Cesme (Turkey), August 2010. (*Relative to Chapter 3*)

- **G. Sanchis-Trilles** and F. Casacuberta. Log-linear weight optimisation via Bayesian Adaptation in Statistical Machine Translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (poster volume), COLING 2010*, pages 1077-1085, Beijing (China), August 2010. (*Relative to Chapter 4*)

There have also been numerous publications indexed in the CORE ranking, but with less estimated impact:

- J. González, **G. Sanchis-Trilles** and F. Casacuberta. Learning Finite State Transducers Using Bilingual Phrases. In *Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2008*, pages 411–422, Lecture Notes in Computer Science, Haifa (Israel), February 2008. (*Relative to Chapter 2*)

- **G. Sanchis-Trilles**, M.T. González, F. Casacuberta, E. Vidal and J. Civera Introducing Additional Input Information into IMT Systems. In *Proceedings of the 5th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, MLMI 2008*, pages 284–295, Utrecht (The Netherlands), September 2008. (*Relative to Chapter 5*)

- **G. Sanchis-Trilles** and M. Cettolo Online Language Model Adaptation via N-gram Mixtures for Statistical Machine Translation. In *Proceedings of the 14th Conference of the European Association for Machine Translation, EAMT 2010*, Saint-Raphaël, (France), May 2010. (*Relative to Chapter 3*)

- **G. Sanchis-Trilles**, D. Ortiz-Martínez, J. González-Rubio, J. González and F. Casacuberta. Bilingual segmentation for phrasetable pruning in Statistical Machine Translation. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation, EAMT 2011*, pages 257–264, Leuven (Belgium), May 2011. (*Relative to Chapter 2*)

Further publications which are neither indexed in the Journal of Citations Report (JCR) ranking nor in the CORE ranking have also been published:

- **G. Sanchis-Trilles** and F. Casacuberta. Increasing Translation Speed in Phrase-Based Models via Suboptimal Segmentation. In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems, PRIS 2008*, pages 135–143, INSTICC Press, Barcelona (Spain), June 2008. (*Relative to Chapter 2*)

- **G. Sanchis-Trilles**, M. Cettolo, N. Bertoldi and M. Federico Online Language Model Adaptation for Spoken Dialog Translation. In *Proceedings of the International Workshop on Spoken Language Translation, IWSLT 2009*, pages 160–167, Tokyo (Japan), December 2009. (*Relative to Chapter 3*)

- N. Bertoldi, A. Bisazza, M. Cettolo, **G. Sanchis-Trilles** and M. Federico FBK @ IWSLT 2009. In *Proceedings of the International Workshop on Spoken Language Translation, IWSLT 2009*, pages 160–167, Tokyo (Japan), December 2009. (*Relative to Chapter 3*)

- V. Alabau, F. Casacuberta, L.A. Leiva, D. Ortiz-Martínez, **G. Sanchis-Trilles**. Sistema web para la traducción automática interactiva. In *Actas del XI Congreso Internacional de Interacción Persona Ordenador, INTERACCION 2010*, pages 47–56, Valencia (Spain), September 2010. (*Relative to Chapter 5*)

- G. Gascó, V. Alabau, J. Andrés–Ferrer, J. González-Rubio, M. A. Rocha, **G. Sanchis-Trilles**, F. Casacuberta, J. González and J. A. Sánchez. ITI-UPV system description for IWSLT 2010 In *Proceedings of the 2010 International Workshop on Spoken Language Translation, IWSLT 2010*, pages 85–92, Paris (France), December 2010. (*Relative to Chapter 4*)

Finally, there is one further publication which has been submitted to an international journal with an estimated impact factor of 2.971, but which has not yet been accepted:

- **G. Sanchis-Trilles** and F. Casacuberta. Batch and online Bayesian predictive adaptation in statistical machine translation. In *Computational Linguistics*. (submitted for revision) (*Relative to Chapter 4*)

In addition, further work carried out during the same period of time than the present thesis, but that is not directly related to the topics presented here, was published in several international conferences and workshops:

- **G. Sanchis-Trilles** and F. Casacuberta. N-Best reordering in Statistical Machine Translation. In *Proceedings of IV Jornadas en Tecnología del Habla, IVJTH*, pages 99–104, Zaragoza (Spain), November 2006.

- **G. Sanchis-Trilles** and F. Casacuberta. Reordering via N-Best Lists for Spanish-Basque Translation. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation, TMI 2007)*, pages 191–198, Skövde (Sweden), September 2007.

- **G. Sanchis-Trilles** and J.A. Sánchez. Vocabulary Extension via POS Information for SMT. In *Proceedings of Mixing Approaches to Machine Translation, MATMT 2008*, pages 63–70, San Sebastián (Spain), February 2008.

- **G. Sanchis-Trilles** and J.A. Sánchez. Using Parsed Corpora for Estimating Stochastic Inversion Transduction Grammars. In *Proceedings of the 6th edition of the International Conference on Language Resources and Evaluation, LREC 2008*, pages 1825–1827 , Marrakech (Morocco), May 2008. (*CORE C*)

- J. González-Rubio, **G. Sanchis-Trilles**, Alfons Juan and F. Casacuberta. A novel alignment model inspired on IBM Model 1. In *Proceedings of the 12th Annual Conference of the European Association for Machine Translation, EAMT 2008*, pages 47–56, Hamburg (Germany), September 2008. (*CORE B*)

- **G. Sanchis-Trilles** and J.A. Sánchez. Phrase segments obtained with Stochastic Inversion Transduction Grammars for Spanish-Basque translation. In *Proceedings of the V Jornadas en Tecnología del Habla, JTH 2008*, pages 119–122 , Bilbao (Spain), November 2008.

- P. Martínez-Gómez, **G. Sanchis-Trilles** and F. Casacuberta. Online learning via dynamic reranking for Computer Assisted Translation. In *Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2011*, pages 93–105 , Tokyo (Japan), February 2011. (*CORE B*)

- P. Martínez-Gómez, **G. Sanchis-Trilles** and F. Casacuberta. Passive-Aggressive for On-line Learning in Statistical Machine Translation. In *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA 2011*, pages 240–247, Las Palmas de Gran Canaria (Spain), June 2011. (*CORE C*)

- G. Gascó, M.A. Rocha, **G. Sanchis-Trilles**, J. Andrés-Ferrer and F. Casacuberta. Does more data always yield better translations?. In *Proceedings of the 13th conference of the european chapter of the Association for Computational Linguistics, EACL 2012, accepted for publication*, Avignon (France), April 2012. (*CORE A*)

6.3 Future work

Research is a never-ending field of work. One never knows where it will end, because it will never end, and the researcher is compelled to keep on pushing the frontier of human knowledge in a constant attempt of breaching it. Hence, once this thesis is completed, a large amount of work remains yet to be done.

Regarding the parameter pruning technique described in Chapter 2, there are two main directions which are worth exploring. The first one concerns the choice of the weighting factor $G(\boldsymbol{y})$. We understand that the choice of $G(\boldsymbol{y})$ is critical, and the goal should be to improve the translation quality obtained by the baseline system, both in statistical machine translation and in interactive machine translation. In this sense, there has been recent work by other research groups (Duan et al., 2011) that points in this same direction, and which shows that there is a research area worth of being explored. The second direction which we intend to explore regards relaxing the different restrictions applied in translation time. Typically, there are several constraints which are applied to the search process so that the computational cost involved is not too high, such as maximum stack size or maximum number of translation options per input phrase. However, given that the techniques presented here reduce such cost, it would be interesting to analyse the effect on translation quality of relaxing such restrictions, given that computational time is not such a big issue after applying the parameter pruning technique described.

Concerning model adaptation, the first step is to apply the adaptation techniques described in this thesis within an interactive machine translation scenario. In this sense, there is already some work being performed, which unveils the adaptation problem in IMT as a problem with its own identity, i.e., techniques that behave correctly in a traditional machine translation setting cannot be applied directly in an interactive scenario. The reason for this is that the metric to be optimised in SMT does not correlate completely with the metric to be optimised in IMT. Although it might seem that this is a minor problem, some adaptation strategies, such as Bayesian predictive adaptation, need to select the best hypothesis in a non-interactive SMT scenario. Moreover, adaptation in IMT might take place even before the full sentence has been validated, and this is bound to open different research possibilities as well.

In Bayesian predictive adaptation, the prior over the model parameters has a key role when computing the best output hypothesis. For this reason, and given the positive results achieved by the implementation presented, we consider that it is a problem which deserves further attention. Furthermore, it can be proved experimentally that no single set of log-linear weights is able to produce the best output, in terms of translation quality, for each one of the input sentences present in the adaptation data. Guided by these two facts, we consider that it would be interesting to consider Gaussian mixtures for the parameter prior.

Given that the four sampling strategies presented yield different performance in terms of final translation quality, another possible extension to the work presented in Chapter 4 consists in studying other possible parameter sampling strategies, as for instance particle filters or other sequential Monte Carlo methods (Doucet et al., 2001).

In addition, we would also like to explore other possible adaptation techniques. One technique that has found a very wide acceptance in speech recognition, but has not been explored as of yet in machine translation, is maximum likelihood linear regression (MLLR) (Christensen, 1998). The application of this technique to machine translation is not straightforward, since the different approximations carried out in the statistical models used in SMT imply that several counts required for the EM estimation may not be computed easily. Nevertheless, we would like to explore this possibility, and analyse to which extent it can be applied to SMT.

Finally, even though in this case the extension proposed in Chapter 5 was performed by only considering the mouse, one could easily imagine different devices which might be transparent to the user and do not necessarily imply overwhelming the user with too many different stimuli, and are yet able to provide the system with very important information. Possible examples might be, for instance, a simple optical pen or even gaze tracking device. In addition, one can also imagine other possible interaction schemes that may take advantage of the mouse (or other devices), and which can boost the productivity of the human translator even further. For example, one such scheme might be to enable the user to select a given part of the translation hypothesis, without requiring that such part must be a specific suffix, and ask the system for other possible translation options for that specific fragment. We plan to research all these possibilities in the near future.

Bibliography

Heidi Christensen. *Speaker Adaptation of Hidden Markov Models using Maximum Likelihood Linear Regression*. PhD thesis, Aalborg University, 1998.

Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001. ISBN 978-1-4419-2887-0.

Nan Duan, Mu Li, Ming Zhou, and Lei Cui. Improving phrase extraction via mbr phrase scoring and pruning. In *Proceedings of the Machine Translation Summit X*, pages 189–197, September 19–23 2011.

List of Figures

1.1 Example of word alignments computed automatically by means of a word-alignment model. The left side shows the alignment as a function of the source sentence (up) and the target sentence (down). On the right side, the alignment is shown in a matricial form, as is often done in SMT. 6

1.2 Example of how consistent phrases are extracted from a word alignment. On the left, the alignment matrix after symmetrisation is shown. Black squares represent word alignments, whereas extracted phrases are marked with a rectangle involving one or more squares. On the right, the phrases that would be extracted from that matrix. Note that word *se* cannot be extracted on its own because its alignment requires word *ha* to be extracted together with it so as to preserve alignment consistency. 10

1.3 Alignment matrix with the different re-ordering types. *m* stands for monotone, *s* stands for swap, and *d* stands for discontinuous. 11

1.4 Example of decoding procedure following the phrases extracted in Figure 1.2, with the input sentence being "`Se ha celebrado en viena una gran conferencia .`". $\kappa_{\boldsymbol{x}}$ illustrates the coverage vector of that specific partial hypothesis. The coverage vector $\kappa_{\boldsymbol{x}}$ of a specific hypothesis keeps track of which words of the source sentence $\boldsymbol{x}$ have been translated until that point, so that words that already have their counterpart in the target sentence $\boldsymbol{y}$ are not translated again. In this figure, character `-` at the n-th position specifies that source word x_n has not been translated yet, and `*` indicates that already has. The probability p of each hypothesis is only for illustrative purposes. 14

1.5 Example of word alignment that results in coverage problems. Maximum phrase length of 7 is assumed. Black squares represent word alignments, whereas extracted phrases are marked with a rectangle involving one or more squares. 15
1.6 IMT session to translate a Spanish sentence into English. Non-validated hypotheses are displayed in italics, whereas accepted prefixes are printed in normal font. 18
1.7 Example of word graph illustrating the translation of *la mesa verde*. $\kappa_{\boldsymbol{x}}$ is the coverage vector of the input sentence (see Section 1.2.1), where symbol `-` indicates an uncovered word, and symbol `*` an input word that has already been translated. Each edge is labelled with both the word emitted when transiting through that edge, and the probability assigned. Note that, for the sake of simplicity, this word graph is not a real example generated during a true search process. 19
1.8 Example of conversion of a phrase graph (left) into a word graph (right). . . . 20

2.1 Amount of phrases present in the reduced system, given as % with respect to the original system. 43
2.2 Decoder speed for original and filtered systems. 43
2.3 Translation quality, as measured by BLEU and TER, for the baseline system and the pruned systems. 44
2.4 Relationship between speed/phrase-table size and the translation quality achieved for the three different strategies analysed. 45
2.5 Relative frequency for each discretized value of $p(\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{x}})$, considering baseline system and different sizes of $G(\boldsymbol{x})$ for the `ModScore` setting (left) and the `QScore` setting (right). 46
2.6 WSR achieved when applying the parameter re-estimation techniques detailed above, for French→English and Spanish→English translation. `flat`, `ModScore` and `Qscore` are the settings defined in the previous sub-section. Confidence intervals are not shown for clarity reasons, but their size was always between 1.64 and 1.90. 47
2.7 Temporal evaluation of the re-estimation techniques detailed above, for French→English and Spanish→English translation. The two plots on the top show how many sentences were produced per second in a user-simulated environment, while the two plots on the bottom show the time consumed in average to produce one single suffix-hypothesis. 48

3.1 Basic procedure for LM adaptation. 57
3.2 Procedure for obtaining development-induced clustering of the training data. 63
3.3 Two-step weight estimation technique. 65
3.4 En→Zh results (BLEU scores and perplexity) for set DEV1 with different grouping methods, sentence specific weight estimation. 67
3.5 En→Zh results (BLEU scores and perplexity) for set DEV2 with different grouping methods, sentence specific weight estimation. 68

3.6 En→Zh results (BLEU scores and perplexity) for set DEV1 with different grouping methods, two-step weight estimation. 69
3.7 En→Zh results (BLEU scores and perplexity) for set DEV2 with different grouping methods, two-step weight estimation. 70
3.8 Zh→En results (BLEU scores and perplexity) for set DEV1 with different grouping methods, sentence specific weight estimation. 71
3.9 Zh→En results (BLEU scores and perplexity) for set DEV2 with different grouping methods, sentence specific weight estimation. 72
3.10 Zh→En results (BLEU scores and perplexity) for set DEV1 with different grouping methods, two-step weight estimation. 73
3.11 Zh→En results (BLEU scores and perplexity) for set DEV2 with different grouping methods, two-step weight estimation. 74

4.1 Algorithm for performing the heuristic sampling described. $\mathrm{rand}(a, b)$ is a value drawn randomly in the interval $[a, b]$, N_s is the desired size of $\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})$ and $\mathbf{s} = [s_1, \ldots, s_k]^T$ is a single sample. 96
4.2 Batch adaptation for different values of delta. News-Commentary corpus considered. In these plots, the size of the n-best list was fixed to 200. 105
4.3 Batch adaptation for different values of delta for the TED corpus. In these plots, the size of the n-best list was fixed to 200. 105
4.4 Translation quality for different variances $\sigma_\mathcal{T}$. In this case, δ was set to 4 and the size of $\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})$ was set to 200. 106
4.5 Batch adaptation with heuristic sampling for different values of δ and n-best. The left plot displays translation quality as measured by TER and the right plot displays confidence interval sizes. The size of the adaptation data was fixed to 100 sentences. Note that, for clarity reasons, the x-axis is *broken*, meaning that the distance between the 1000 and 10000 ticks is actually altered. 107
4.6 Effect of increasing the size of $\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})$, i.e., $|\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})|$ denoted by nw in the plot, on the size of the confidence intervals. δ was set to 4, and the size of the n-best list to 200. 108
4.7 Batch adaptation with Viterbi sampling and different amount of adaptation set sizes. The size of the n-best list was fixed to 200 and $\sigma_\mathcal{T} = 0.1$. 108
4.8 Gaussian sampling strategy for batch adaptation of $\boldsymbol{\lambda}$. 109
4.9 Translation quality for different values of $\sigma_\mathcal{T}$ and σ_o within the MCMC procedure for BPA. $\delta = 1$ and size of n-best set to 200. 110
4.10 Effect of considering different burn-in durations, when varying the MCMC chain length. The number of sampled weights after the burn-in phase (i.e., $|\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})|$) is denoted by n. The number of adaptation samples was set to 100. $\sigma_\mathcal{T} = 0.1$ and $\sigma_o = 0.01$. 111
4.11 Effect on translation quality and confidence interval sizes of considering different $\mathcal{S}(\boldsymbol{\theta}_\mathcal{T})$ sizes, denoted by nw in the plot. Burn-in duration was set to 200. $\sigma_\mathcal{T} = 0.1$ and $\sigma_o = 0.01$. 111
4.12 Translation quality for batch adaptation with MCMC sampling and different sizes of n-best, represented as N in the plot. $\sigma_\mathcal{T} = 0.1$ and $\sigma_o = 0.01$. 112

4.13 Translation quality, as measured by TER, obtained when comparing performing a full re-estimation of $\boldsymbol{\lambda}$ by means of MERT, and when using the same adaptation data as adaptation set within BPA. News-Commentary corpus considered. 113

4.14 Translation quality, as measured by TER, obtained when comparing performing a full re-estimation of $\boldsymbol{\lambda}$ by means of MERT, and when using the same adaptation data as adaptation set within BPA. TED corpus considered. 114

4.15 Translation quality, as measured by BLEU, obtained when comparing performing a full re-estimation of $\boldsymbol{\lambda}$ by means of MERT, and when using the same adaptation data as adaptation set within BPA.TED corpus considered. . 114

4.16 Time in minutes consumed by the different adaptation approaches compared. In the case of BPA, 2000 samples for $\boldsymbol{\lambda}$ were obtained, and the size of the n-best list was set to 200. Note that both axes are shown in logarithmic scale. News-Commentary corpus considered. 115

4.17 Effect of different δ leveraging factors in online adaptation. The size of the n-best list was fixed to 200. $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 2000$ and $\sigma_{\mathcal{T}} = 0.1$. 117

4.18 Confidence interval sizes for different sizes of $\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})$, denoted by `nw` in the plot. 117

4.19 Translation quality when considering increasing n-best size. $\delta = 2$, $\sigma_{\mathcal{T}} = 0.1$ and $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 2000$. 118

4.20 Effect of varying $\sigma_{\mathcal{T}}$ within $\mathcal{N}(\boldsymbol{\lambda}_{\mathcal{T}}, I \cdot \sigma_{\mathcal{T}})$. The size of the n-best list was set to 200. $|\mathcal{S}(\boldsymbol{\theta}_{\mathcal{T}})| = 2000$ and $\delta = 2$. 119

4.21 Translation quality and confidence interval sizes when using BPA as a stabilisation method. `baseline` computed by means of MERT, `mert-BPA` stands for the BPA approach when using the MERT-weights as mean vector within the BPA prior, and `euro-BPA` stands for the BPA approach when using the Europarl weights within the BPA prior. Both plots present the x-axis in log-scale and, in addition, the `CFI` plot also presents the y-axis in log-scale. The size of the training data is given in thousands of sentences. . . 120

5.1 Example of anticipated proposal pointer action which solves an error of a missing word. In this case, the system produces the correct suffix $\boldsymbol{s}_h$ immediately after the user validates a prefix $\boldsymbol{p}$, implicitly indicating that we wants the suffix to be changed, without need of any further action. In **ITER-1**, character | indicates the position where a pointer action was performed, $\boldsymbol{s}_l$ is the suffix which was rejected by that pointer action, and $\hat{\boldsymbol{s}}_h$ is the new suffix that the system suggests after observing that $\boldsymbol{s}_l$ is to be considered incorrect. Character # is a special character introduced by the user to indicate that the hypothesis is to be accepted. 130

5.2 Example of partial refusal pointer action which corrects an erroneous suffix. In this case, an anticipated proposal pointer action is performed in **ITER-1** with no success. Hence, the user introduces word "*type*" in **ITER-2**, which leaves the cursor position located immediately after word "*type*". In this situation the user would not need to perform a pointer action to re-position the cursor and continue typing in order to further correct the remaining errors, since he could simply continue typing the word he has in mind. However, since he has learnt the potential benefit of pointer actions, he performs a partial refusal pointer action in order to ask for a new suffix hypothesis, which happens to correct the error. 132

5.3 WSR improvement when considering one to five maximum PAs. All figures are given in %. The left column lists WSR improvement versus PAR degradation, and the right column lists WSR improvement versus uPAR. Confidence intervals at 95% confidence level following (Koehn, 2004). 135

List of Tables

1.1 Characteristics of Europarl for each of the sub-corpora. OoV stands for "Out of Vocabulary" words with respect to (wrt) the specified training corpus. Devel. stands for Development, k for thousands of elements and M for millions of elements. 22

1.2 Characteristics of the three News-Commentary test sets that will be used. `Training` refers to the News-Commentary training set. OoV stands for "Out of Vocabulary" words with respect to (wrt) the specified training corpus. NC stands for News-Commentary, k for thousands of elements and M for millions of elements. 23

2.1 Translation quality, number of model parameters measured in terms of millions of phrase pairs, number of translated words per second and speedup (S_p) obtained when using a PB translation system for the source-driven segmentation technique. Monotonic search was considered. PB model size is given in millions of phrase-pairs. 41

2.2 Translation quality, number of model parameters, number of translated words per second and speedup (S_p) obtained when using a PB translation system for the true segmentation technique. Monotonic search was considered. PB model size is given in millions of phrase-pairs. 42

3.1 Statistics of the IWSLT training data. |W| stands for running words, |V| for vocabulary size and $\bar{s}$ for average sentence length. 59

3.2 Speaker-based statistics of the CT training set. 60

3.3 English side statistics of the Nespole! dialogues. 61

3.4 Most frequent Nespole! dialogue acts. 61

3.5 Speaker-based statistics of the CT development set. 66

3.6 MERT effect on the BLEU score. 72
3.7 Performance of the direct clustering approach. 75
3.8 Performance of the development-induced clustering approach. The best results are marked in bold. 76
3.9 Performance of the test-induced clustering approach. 77

4.1 Main figures of the News-Commentary corpus. *OoV* stands for Out of Vocabulary. k/M stands for thousands/millions of elements. 103
4.2 Main figures of the TED corpus. *OoV* stands for Out of Vocabulary. k/M stands for thousands/millions of elements. 103
4.3 Analysis of n-gram precision and brevity penalty for 10 and 100 adaptation samples, considering heuristic and MCMC sampling within BPA and MERT and MERT+ strategies (i.e., including the adaptation data only, or the adaptation and development data for re-estimating $\boldsymbol{\lambda}$. NC corpus considered. 115

5.1 WSR improvement when considering non-explicit MAs. "rel." indicates the relative improvement. All results are given in %. 134