Université du Maine

Model Adaptation techniques in Machine Translation

THÈSE

présentée et soutenue publiquement le 29 Juin 2012

pour l'obtention du

Doctorat de l'Université du Maine

(spécialité informatique)

par

Kashif Shah

Composition du jury

Rapporteurs :	Prof. Kamel Smaïli	Professeur	LORIA, Université de Nancy 2
	Prof. Philippe Langlais	Professeur	RALI, Université de Montréal
Examinateurs :	Prof. Laurent Besacier	Professeur	LIG, Université J. Fourier
Directeur de thèse :	Prof. Holger Schwenk	Professeur	LIUM, Université du Maine
Co-encadrant :	Dr. Loïc Barrault	Maître de Conférence	LIUM, Université du Maine

Laboratoire d'Informatique de l'Université du Maine

Dedicated to Pakistan.

Acknowledgements

All praise is due to Allah who gave me the strength, understanding, patience, ability and help to accomplish this work.

Foremost, I would like to express my sincere gratitude to my advisor Prof. Holger Schwenk for the continuous support of my PhD study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined to finnish my PhD without his continuous supervision.

My sincere thanks also goes to Dr. Loïc Barrault for his co-supervision. His wide knowledge and his logical way of thinking have been of great value for me. His understanding, encouraging and personal guidance have provided a good basis for the present thesis.

Besides my advisors, I would like to thank my thesis committee: Prof. Laurent Besacier, Prof. Philippe Langlais and Porf. Kamel Smaïli for their encouragement, insightful comments and questions. I am truly grateful to all my jury members for the wonderful defense experience they gave me.

I am deeply thankful to the Higher Education Commission of Pakistan (HEC) for providing research grant under Overseas Scholarship 2008, the French government under the project COSMAT (ANR 2009 CORD 004 01), and the European project FP7 Euromatrix Plus.

I thank my fellow labmates in Language and Speech Technology group (LST): Patrik Lambert, Christophe Servan, Mohammed Attik, Sadaf Abdul-Rauf, Anthony Rousseau, Frédéric Blain, Haithem Afli, Walid Aransa, Rahman Ali and Huei-chi Lin, for the stimulating discussions, for the late evenings we were working together before deadlines, and for all the fun we have had in the last three years. I also would like to thank Fethi Bougares, Grégor Dupuy, Richard Dufour, Antoine Laurent, Aina Lekira, Carole Lailler, Elie Khoury, Vincent Jousse, Thierry Bazillon and Paul Gay for their nice company.

Special thanks to Patrik Lambert for his great assistance and guidance in many research problems; Haithem Afli, Fethi Bougares, Rahman Ali and Huei-chi Lin for inviting me often to eat together and sharing their food specialties; Frédéric Blain and Grégor Dupuy for their help in french language stuff and Sadaf Abdul-Rauf for important tips during thesis writing.

Members of Computer science laboratory (LIUM) also deserve my sincerest thanks, their friendship and assistance has meant more to me than I could ever express. I am specially grateful to friendly assistance of kind professors: Dominique Py, Yannick Estéve, Sylvain Meignier, Paul Deléglise and Nathalie Camelin. I would also like to acknowledge the support and assistance of administrative and technical staff: Martine Turmeau, Etienne Micoulaut, Teva Merlin and Bruno Richard.

My friends in France, Pakistan and other parts of the World were sources of laughter, joy, and support. I am very happy that, in many cases, my friendships with them have extended well beyond our shared time.

Last but not the least, I would like to thank my family: my brothers, sister, cousins, aunt, uncle and above all my parents. Their love provided my inspiration and was my driving force. I owe them everything and wish I could show them just how much I love and appreciate them. I would like to dedicate my work specially to my father (May God rest his soul in peace) for supporting me spiritually throughout my life and provided me the sound platform to build my carrier. He walks beside us everyday, unseen and unheard but always near. I hope that this work makes him proud.

Abstract

Nowadays several indicators suggest that the statistical approach to machine translation is the most promising. It allows fast development of systems for any language pair provided that sufficient training data is available. Statistical Machine Translation (SMT) systems use parallel texts - also called bitexts - as training material for creation of the translation model and monolingual corpora for target language modeling. The performance of an SMT system heavily depends upon the quality and quantity of available data. In order to train the translation model, the parallel texts is collected from various sources and domains. These corpora are usually concatenated, word alignments are calculated and phrases are extracted. However, parallel data is quite inhomogeneous in many practical applications with respect to several factors like data source, alignment quality, appropriateness to the task, etc. This means that the corpora are not weighted according to their importance to the domain of the translation task. Therefore, it is the domain of the training resources that influences the translations that are selected among several choices. This is in contrast to the training of the language model for which well known techniques are used to weight the various sources of texts.

We have proposed novel methods to automatically weight the heterogeneous data to adapt the translation model. In a first approach, this is achieved with a resampling technique. A weight to each bitexts is assigned to select the proportion of data from that corpus. The alignments coming from each bitexts are resampled based on these weights. The weights of the corpora are directly optimized on the development data using a numerical method. Moreover, an alignment score of each aligned sentence pair is used as confidence measurement.

In an extended work, we obtain such a weighting by resampling alignments using weights that decrease with the temporal distance of bitexts to the test set. By these means, we can use all the available bitexts and still put an emphasis on the most recent one. The main idea of our approach is to use a parametric form or meta-weights for the weighting of the different parts of the bitexts. This ensures that our approach has only few parameters to optimize.

In another work, we have proposed a generic framework which takes into account the corpus and sentence level ”goodness scores” during the calculation of the phrase-table which results into better distribution of probability mass of the individual phrase pairs.

We have reported the experimental results for various reputed international evaluation tasks including IWSLT, NIST OpenMT, and WMT on English-French/Arabic language pairs and showed significant improvements in translation quality.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the beginning of the human kind, the diversity of the cultures, the traditions and the languages have changed immensely along different geographical boundaries. It is widely acknowledged that we are living in an era in which global communication technology has enabled us to go beyond borders and that we are observing the beginning of a global community. Given the fact that many obstacles are sorted out to create the a global village in the world, there are still certain barriers which require to be jumped over. The language barrier is considered as one of the main hurdle for global communication. Each community developed social and scientific literature in their own languages, which is understandable to their own community and hence profitable to a limited group of people. The need to know and understand the material published in other languages compelled people for text translation. It gave an immense need to hire bilingual or multilingual translators. But the text in various languages was so enormous that it seemed impossible for human translators to translate this huge amount of data.

The need for machine translation has emerged as possible alternative to human translators. Specially, revolutionary development in computer technology and their capability to process mountains of data made it possible to develop reasonable translation systems. In this world of global communication, machine translation has become a key technology. We can anticipate that translation is an essential tool in our daily life.

The idea of machine translation may be traced back to the 17th century when René Descartes proposed a universal language, with equivalent ideas in different languages sharing a common symbol. During 1950s, the Georgetown experiment (1954) involved fully automatic translation of several Russian sentences into English. The authors claimed that within three to five years machine translation would be a solved problem. However the actual progress was much slower. After the ALPAC report (1966), which found that the many-year-long research had failed to fulfill expectations, funding was greatly reduced. During the late 1980s, as computational power increased and became less expensive, more interest went to statistical models for machine translation. Currently, the scientific community around this topic is growing rapidly, the performance of such systems are constantly improving and large companies such as Google, Microsoft and IBM are investing heavily in this area.

The automatic translation of texts is a research topic for several decades and various approaches have been proposed. They initially were based on linguistic rules. Generally, rule-based approaches parse a text, usually creating an intermediate, symbolic representation, from which the text in the target language is generated. Based on an intermediate representation, an approach is described as interlingual machine translation or transfer-based machine translation. These methods require large lexicons with morphological, syntactic, and semantic information, along with large sets of rules. On the other hand, data-driven machine translation systems often work well enough for a native speaker of one natural language to get an approximate translation of what is written in another language provided that sufficient data is available. The arguments in support of a particular method are orthogonal. For example, the large multilingual corpus of data needed for statistical methods is not necessary for the grammar-based methods. But then, the grammar based methods need a skilled linguist to carefully design the grammar between the various language pairs.

Now several indicators suggest that the statistical approach to machine translation is the most promising. It allows fast development of systems for any language pair, provided that sufficient training data is available. Statistical Machine Translation (SMT) systems use parallel texts – also called bitexts – as training material for creation of the translation model and monolingual corpora for target language modeling.

While monolingual texts are in general easily available in many domains, the freely available parallel texts mainly come from international organisations, like the European Union or the United Nations. These texts, written in a particular jargon, are usually much larger than in-domain bitexts. As an example, we can cite the development of Arabic/English translation system with data available in OpenMT evaluation by NIST[1]. The current NIST test sets are composed of a news wire part and a second part of web-style texts. For both domains, there are only several millions of words of in-domain bitexts available, in comparison to almost 200 millions words of out-of-domain United Nation (UN) texts. The later corpus is therefore likely to dominate the estimation of the probability distributions of the translation model.

Performance of statistical machine translation heavily depends upon quality and quantity of available data. Today, most SMT systems are generic, *i.e.* the same system is used to translate texts of all kinds. Therefore, it is the domain of the training resources that influences the translations that are selected among several choices.

The parallel data is (wrongly) considered as one homogeneous pool of knowledge. It is argued that the parallel data is quite inhomogeneous in many practical applications with respect to several factors:

- the data may come from different sources that are more or less relevant to the translation task (in-domain versus out-of-domain data).
- more generally, the topic or genre of the data may be more or less relevant.
- the data may be of different quality (carefully performed human translations versus automatically crawled and aligned data).
- the recency of the data with respect to the task may have an influence. This is of interest in the news domain where named entities, etc change over time.

These factors may prevent good domain specific translations. For instance, the English word ”cluster” may be translated into the French word *grappe* (informatics), *regroupement* (mathematics), *amas* (astronomy), etc.

[1] National Institute of Standard and Technology

These problems could be tackled by adapting the models to the given domain. Current state-of-the-art SMT systems are based on many models while domain adaptation is centered around two: the translation model (TM) and the language model (LM). While many techniques exist to adapt a language model to a specific domain (most of them are borrowed from automatic speech recognition), it seems that very little research has been done that seeks to apply similar ideas to the translation model. The work presented in this thesis is centered on translation model adaptation when the bitexts are heterogeneous with respect to the above mentioned factors.

1.1 Scientific goals and objectives

The aim of this PhD thesis is to perform domain adaptation of state-of-the-art SMT systems by employing new approaches which are flexible, efficient and robust. In particular, the following scientific goals are pursued:

- to focus on **translation model adaptation**. As described above, there are well known techniques to adapt language models, borrowed from speech recognition, but there doesn't exist well recognized approaches to adapt the translation model. Therefore, the work presented in this thesis is focused on translation model adaptation.

- to **exploit given parallel corpora as much as possible**. A straightforward way to adapt a model is to collect more domain specific data. This is conceivable for monolingual data, but quite difficult and costly for bilingual data, as already mentioned. Therefore, alternative approaches were proposed in the literature: extracting bitexts from comparable corpora, *e.g* [Abdul-Rauf and Schwenk, 2009; Do, 2011; Munteanu and Marcu, 2005] or unsupervised training, *e.g* [Schwenk and Senellart, 2009]. In this work, we do not use any additional data, but we strive to take best advantage of existing resources.

- to use **training data which is freely and easily available**. This is an important constraint because of the fact that the SMT systems heavily depend upon the training resources used to built those models. By this mean the systems developed at different labs by various researchers are comparable easily. The data used in our experiments is provided by international organizations such as Europarl, News-Commentary, UN etc and is freely available. Therefore, anyone could employ the proposed techniques and reproduce the results with ease.

- to consider **an automatic way of adapting** the models to the domain of interest. Sometimes, the domain of the training data is unknown and often even a training corpora may contain the text from various domains and the quality of the data might be questionable. An automatic way of adapting the models is considered in our experiments.
- to work on a **language independent techniques** which could be employed on any language pair. In this thesis, the proposed techniques are generic and the do not depend on the language pairs.
- to show the **improvements in translation quality on very competitive systems** which are well ranked in various internationally evaluation campaigns. During this PhD thesis, the experiments are done on state-of-the-art systems built at LIUM. These systems were among the best systems in well known international evaluation campaigns, namely NIST OpenMT, WMT and IWSLT in the period 2009 to 2012.
- to propose a **generic approach** which gives the flexibility to inject alternative features that estimate the quality and appropriateness of heterogeneous corpus.

1.2 Research Contribution

The work done in this thesis is based on the following assumptions:

- The parallel training data is heterogeneous with respect to genre, source, topic, size and quality.
- The domain of the data is unknown.

- We dispose of state-of-the-art SMT systems for an arbitrary language pair.

Based on those assumptions the contributions of this thesis are as follows:

- We propose to **resample** the bitexts coming from different sources. Considering the fact that the performance of the SMT system is proportional to the quantity of training data used to build those systems, data is collected and merged together regardless of its source, nature/domain, size and quality. This introduces the sample bias problem. Any statistics calculated from the biased sample are erroneous and can lead to under or over representation of related parameters. In other words, it will not accurately represent the target domain. The sample bias is adjusted by picking the random sample, which results into reasonable approximations of related parameters. A weight to each sample - also called sample weight - is introduced for appropriate estimation from randomly selected sample drawn from complete sample. In our experiments we considered bitexts from each source as a different sample. A weighting coefficient to each bitexts is associated. This method does not require an explicit specification of the in-domain and out-of-domain training data. The weights of the corpora are directly optimized on the development data using a numerical method, similar to the techniques used in the standard minimum error training of the weights of the feature functions in the log-linear criterion. This work has been published in Shah et al. [2010].

- In another approach, we exploited the so-called **recency effect** to weight the bitexts. In principle, the data with less temporal distance will most likely be similar with respect to style and vocabulary. The recency effect is of interest specially in the news domain where named entities, etc change over time. The idea is to consider some kind of meta-weights on the continuous stream of training data over time. Instead of dividing the corpora in different parts and numerically optimizing all the weights for each part, these meta-weights only depend on few parameters that need to be optimized. The weighting is still done by resampling. This work has been published in Shah et al. [2011]

- In a third approach, we integrate various weighting schemes directly in the calculation of the translation probabilities. Resampling the bitexts or alignments is computationally expensive for large corpora since the resampled data is ideally much bigger than the original one. Instead, we directly worked on the translation probabilities. This work is an extension and generalization of several ideas proposed in previous works such as weighted counts with feature scores. However, our proposed framework gives the flexibility to inject the feature scores in a unified formulation calculated at various levels. It is based on the following principles:
    - the use of a set of "quality measures" at different levels: weights for each corpus (or data source) and for each individual sentence in the bitexts.
    - a small number of parameters to be optimized.
    - we do not introduce additional feature functions during decoding, but we only modify the existing phrase probabilities. By these means, we don't have to deal with the additional complexity of decoding and optimizing the weights of many feature functions.

1.3 Outline of the thesis

The thesis is organized into 6 chapters.

Chapter 2 summarizes the current state-of-art SMT technology relevant to our work. A brief description of various MT approaches is presented first. Then, we describe the mathematical framework of early word-based SMT systems and present phrase-based systems as a natural evolution of the original approaches. An overview of the word alignment process and the different alignment models is also given. Language modeling concepts are also presented. The section on SMT concludes by discussing the difficulties of MT evaluation and presenting the most commonly used evaluation metrics. In the next section, the need for model adaptation is explained and we give a detailed analysis of approaches proposed in the literature.

Chapter 3 describes our approach to adapt translation models by resampling. We start by a description of the algorithm, followed by the general architecture and we then present the individual components in detail. The approach is evaluated on two well known evaluation tasks, *i.e.* NIST OpenMT and IWSLT.

Chapter 4 presents an extended work to weight the corpora by using parametric weighting. We first describe the algorithm along with the architecture of our proposed approach. We then present the experimental evaluation on the WMT task showing improvements over our baseline systems. At the end, we discuss our approach by showing various examples along with their translations.

Chapter 5 finally explains a general framework to weight the parallel data on various levels by an efficient approach, along with various methods to calculate the corpus weights and sentence features. Experiments on the WMT and IWSLT tasks are reported and a comparative analysis of the impact on the translation quality with various weights and features is presented.

Chapter 6 concludes the thesis by presenting a brief summary of the work and discussing various prospects for future research.

Chapter 2

Survey of Machine Translation

In this chapter we start with the introduction of Machine Translation (MT) followed by a brief overview of various approaches and concepts applied to MT. Subsequently, main ideas of SMT along with an explanation of fundamental techniques are presented. The experiments and results reported in this thesis are based on the so-called phrase-based approach, therefore we will discuss in detail its underlying concepts and ideas.

2.1 Machine translation

Machine translation is the process of translating automatically from one natural language into another. There are some close terms being used in literature such as machine-aided translation or computer-assisted translation. These terminologies are generally used in the context where computer programs are employed to assist or to facilitate human translators.

There are various strategies to perform machine translation, which are inspired by the principles how languages are analyzed – from complete linguistic representation to direct translation. Generally, machine translation systems are categorized in the following three approaches as shown in the well-known machine translation pyramid of Bernard Vauquois (Figure 2.1):

- Direct approach
- Transfer-based approach

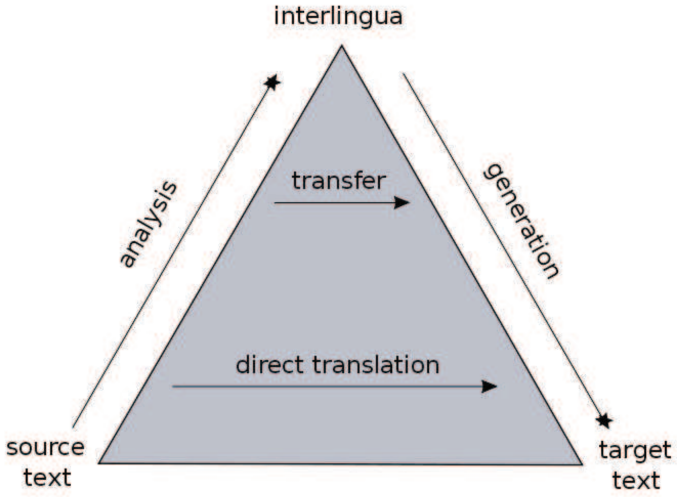

Figure 2.1: Bernard Vauquois' pyramid

- Interlingua

Figure 2.2 shows the analysis performed at each level in the pyramid. As we climb up the pyramid deeper analysis is performed. A brief overview of each approach is given in next the section. A more detailed description can be found in [Hutchins and Somers, 1992].

2.1.1 Direct approach

The direct translation approach belongs to the first generation of translation systems in which source text is directly translated into target text without any intermediate steps. This approach was first introduced in the 1950s, the computer systems were thousand times less powerful as compared to the present systems. No high level language was available and most of the programming was done in assembly language. The approach was based on a bilingual dictionary look-up to translate the text. The translation unit of the approach was usually a word. A direct translation system is developed for a specific source and target language pair. No syntactic or semantic analysis is performed during the translation process.

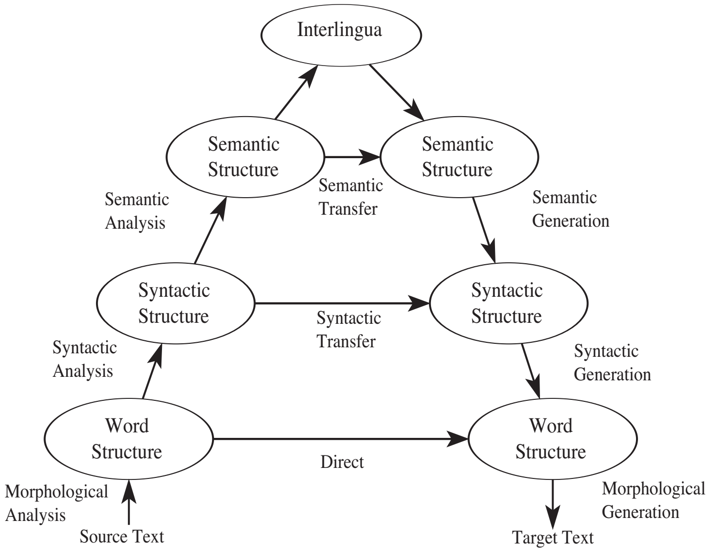

Figure 2.2: Analysis performed in machine translation pyramid

2.1.2 Transfer-based approach

The first generation translation systems failed to perform well in the absence of syntactic and semantic analysis and due to limited computational resources. It was soon proposed to use linguistic models in the translation process to some extent, or to create an intermediate representation of the source languages that could be used for translation into the target language. This gave birth to indirect approaches like the transfer-based-approach and interlingua.

The transfer-based approach is based on the analysis of the text in the source language to form an intermediate representation which is then used to generate text in the target language. Generally, three main modules are involved in this approach, *i.e.* analysis, transfer and generation [Arnold et al., 1993]. The intermediate representations are language dependent and for any language pair there is an unique intermediate representation which could be used to perform translation between the given language pairs. In multilingual system, the addition of a new language pair involves not only the development of a separate module for text analysis and generation, but also an new transfer module between them. So, for n languages there would be n analysis, n generation and n(n-1) transfer modules. This was a clear drawback of the approach which led the researchers to look for a new framework in which the addition of a new language requires less work.

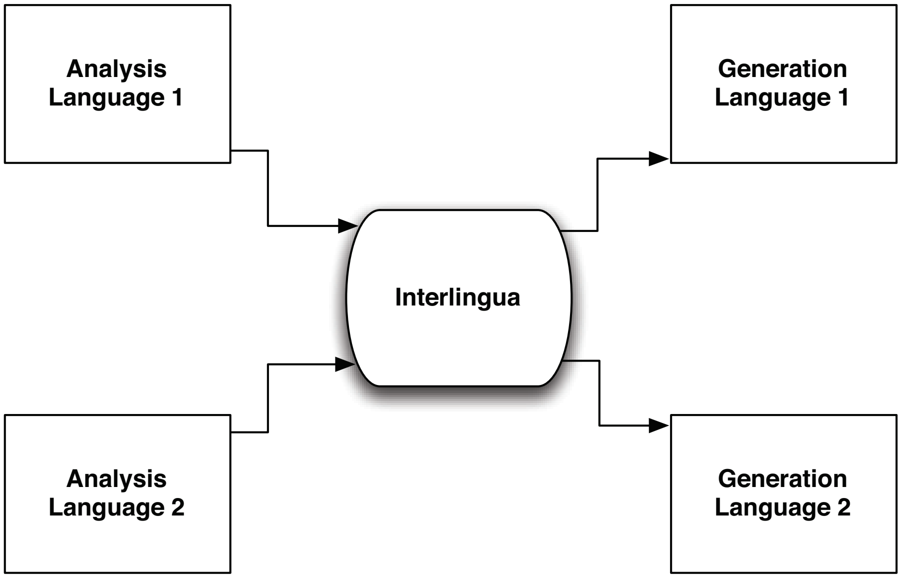

Figure 2.3: Interlingua machine translation

2.1.3 Interlingua

Interlingua translation systems are based on an abstract language-independent approach called interlingua. The source text is converted into interlingua which is then used to generate target language as shown in figure 2.3. The most difficult module of this approach is to create an interlingua that would be language independent [Hutchins and Somers, 1992]. It may be possible in an ideal scenario to create such an interlingua for a specific domain in a multilingual system. The obvious advantage is to create fewer new modules when a new language is added into the multilingual system, *i.e.* in contrast to the transfer-based approach, it requires only 2n modules.

2.2 Corpus-based approach

With the availability of reasonable amounts of already human-translated data in various languages, it was realized that the translations could be generated with ease and in less time consuming manner by using translation memories which contain multilingual corpora.

The corpus-based approach is also termed as an empirical approach in which the knowledge sources to develop an MT system are computed automatically by analyzing example translations. A major benefit of empirical approaches is that MT systems for new language pairs and domains can be developed rapidly and with ease, provided sufficient training data is available. Figure 2.4 shows the architecture of a corpus-based MT system. Generally, the starting point is a parallel training corpus consisting of translation examples produced by human translators. In the training phase, the necessary knowledge base is computed automatically. The search or decision process is employed to achieve an optimal combination of the knowledge sources to perform the best possible translation. Corpus-based approaches are based on sentence-aligned parallel corpora and good alignment is important for better translation output. A good review of algorithms that have achieved good results for sentence alignment is described in [Gale and Church, 1993; Haruno and Yamazaki, 1996; Kay and Röscheisen, 1993]. The two main corpus-based approaches which have emerged are example-based approach and statistical-based approach.

In the example-based approach, translation is performed by analogy. The source sentence to be translated is searched in parallel corpora, translation examples are extracted and combined to generate the target sentence. Somers [1999] gives a survey of various EBMT techniques, whereas Hutchins [2005] presents a historical review.

Statistical machine translation is based on statistical models which are trained on bilingual and monolingual data. Today, a lot of work is being done to include in addition syntactic and linguistic information. In the beginning, the approach was based on words, and therefore, it could be classified as a direct approach. In the following section, various models and techniques used in SMT are discussed thoroughly.

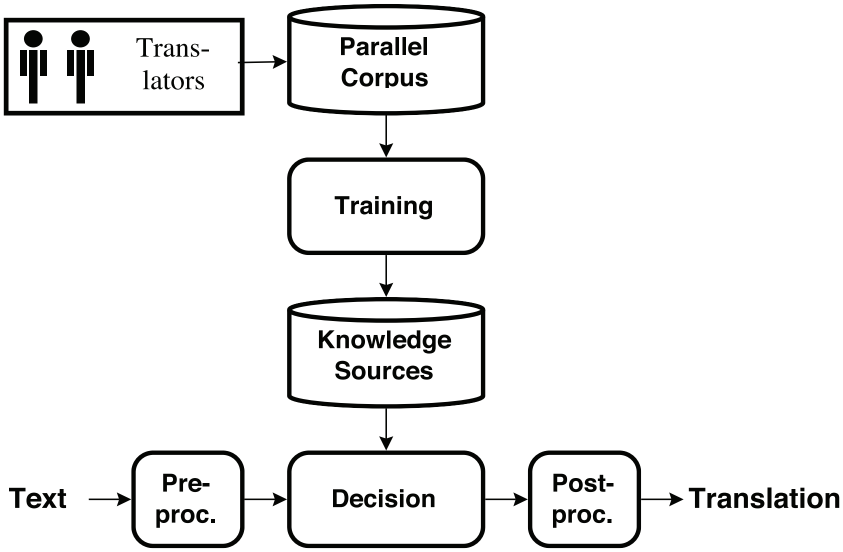

Figure 2.4: Corpus-based machine translation (taken from [Och, 2002])

2.3 Statistical machine translation

Modern machines can be programmed to learn from empirical data with the design and development of sophisticated algorithms. The algorithms automatically extract knowledge to recognize complex patterns and make intelligent guesses based on data. This paradigm is named machine learning and is widely used to solve many problems. SMT is based on translation by learning statistical models. These models are learned by already available translated text.

First, we formally describe the task of translating a sentence f in the source language to a sentence e in the target language. Treating sentences as sequences of words, we define them as $f = f_1, ..., f_i, ..., f_I$ and $e = e_1, ..., e_j, ..., e_J$, where f_i and e_j denote the words in position i of f and position j of e, respectively. In a historical prospective, SMT is viewed as a noisy channel paradigm, which considers the translation process as a channel which distorts the target sentence and outputs the source sentence. It is derived from the Bayes' theorem which is written as:

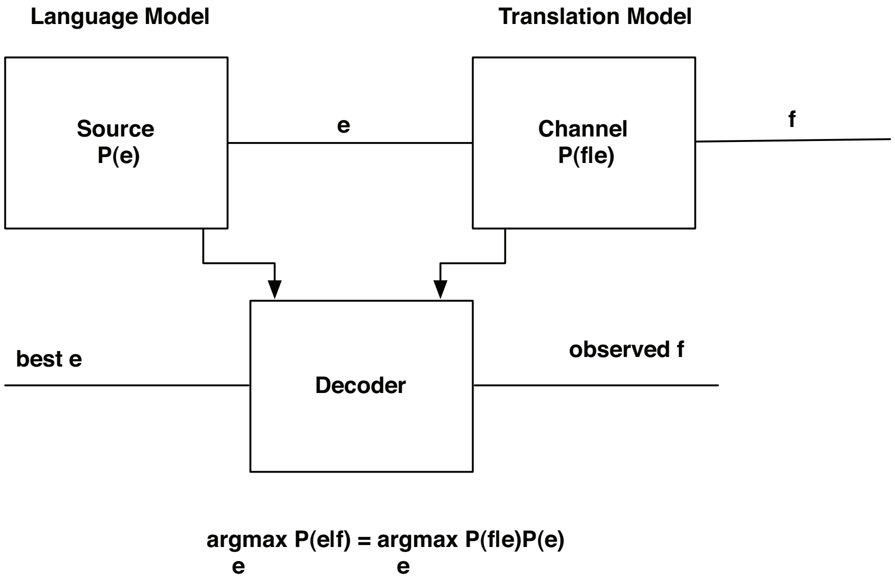

Figure 2.5: SMT based on source-channel models

$$e^* = \arg\max_e Pr(e|f) = \frac{Pr(e)Pr(f|e)}{Pr(f)} \tag{2.1}$$

Independence of the denominator $Pr(f)$ and our goal to maximize the product simplifies equation 2.1 to

$$e^* = \arg\max_e Pr(e)Pr(f|e) \tag{2.2}$$

Equation 2.2 is the fundamental equation of statistical machine translation and divides the overall framework into two separate models: a target language model $Pr(e)$ and a translation model $Pr(f|e)$. The translation process consists in maximizing the product of the two functions. The translation model (TM) is learned from parallel corpora whereas the language model (LM) is build on monolingual data. The translation model provides the best translation according to the input text while the target language model ensures that the translation is grammatically correct, regardless of the input text. The **decoder** performs the search for the best translation e^* given the search space of all possible translations based on the probabilities of the language and translation models.

There exists numerous techniques to represent and train translation models. They vary in the choice of the basic translation unit. Early translation models were based on a word-by-word alignment model and word translation probabilities. Recent systems operate on sequences of words, called phrase-based systems. A phrase is a contiguous sequence of words and a phrase pair is a translation equivalent of each other in a given language pair. These phrase pairs are stored along with their frequency and are then used as building blocks for new translations. Also some phrase-based SMT approaches are based on non-contiguous phrases or phrases with gaps as in [Gimpel and Smith, 2011; Simard et al., 2005]. The probability distributions of all these statistical models are automatically estimated from a sentence-aligned parallel corpus.

2.3.1 IBM translation models

It is not practical to model full sentences since most sentences occur only once or few times even in large texts. Therefore, breaking up the sentences into smaller components is an ultimate choice, *e.g.* words which are more auspicious to collect enough statistics to estimate probability distribution. [Brown et al., 1993, 1990] proposed word-based models in the 1990s which opened the door for the researchers in many domains. These models are based on the same noisy channel. According to equation 2.2, we need to calculate the *inverted* translation probability $Pr(f|e)$ in order to translate the text f in the source language to a string e in the target language. The first difficult task is to establish the correspondences between the words in source and target sentences. Typically, the number of words and the order of the corresponding appearances in translated sentences is different. This problem is handled by using a hidden variable a which accounts for all possible pair-wise alignment links between the two sentences:

$$Pr(f|e) = \sum_a Pr(f, a|e) \tag{2.3}$$

$Pr(f, a|e)$ can be expanded to smaller models as

$$Pr(f,a|e) = Pr(J|e)\prod_{j=1}^{J} Pr(a_j|a_1^{j-1}, f_1^{j-1}, J, e)Pr(f_j|a_1^{j}, f_1^{j-1}, J, e) \tag{2.4}$$

where,
J = length of the source sentence f
f_j = word in position j of source sentence f
a_j= hidden alignment of word f_j indicating the position at which f_j aligns in the target sentence

which says that given target sentence f, choose the length J of source sentence e from which we could find where to link the first source position given target sentence and the length of source sentence. Then given target sentence, the length of source sentence and the target word linked to the first source position, we could identify the first source word and so on. The source words that are not aligned to any target words are aligned to empty word (NULL). The Expectation-Maximization (EM) algorithm is used to find out hidden parameters by maximizing the likelihood of the parallel corpus. There are 5 models proposed by IBM.

Model 1 makes use of co-occurrence of word pairs and is based on lexical translation probability distribution only. The translation probability between given foreign sentence $f = (f_1, f_2, ..., f_{lf})$ and target sentence $e = (e_1, e_2, ..., e_{le})$ with an alignment of each target word e_j to a foreign word f_i according to the alignment function a could be written as:

$$p(e,a|f) = \frac{\epsilon}{(l_f+1)^{l_e}}\prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \tag{2.5}$$

which shows the generated output words e_j for all l_e by taking the product over all the lexical translation probabilities along with normalization. ϵ is a normalization constant to make the summation of all possible target translation probabilities to one. The lexical translation probabilities are learned by EM algorithm as follows:

- Initialize the model with uniform distribution
- Expectation step by applying the model over the data
- Maximization step by learning the model over the data
- Loop through step 2 and 3 unless converged

The convergence of EM algorithm is determined by calculating the perplexity of the model at each iteration. The perplexity is calculated as follows:

$$\log_2 PP = -\sum_s \log_2 p(e_s|f_s) \tag{2.6}$$

The perplexity is supposed to decrease at each iteration and it ensures the convergence of the EM algorithm.

IBM model 1 gives the basic functionality to build word-to-word alignments, but it has many defects. For example, it doesn't take care about word reordering.

IBM model 2 introduce the notion of alignment probability distribution based on the positions of the words. The two steps to build IBM model 2 involving lexical translation step and alignment step are formulated as:

$$p(e,a|f) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)})a(a(j)|j,l_e,l_f) \tag{2.7}$$

It could be observed from the above equation that IBM model 1 is a special case of IBM model 2 where second part of the product is fixed in model 1. Though IBM model 2 seems reasonable to tackle with the problem of reordering but still things are more complicated when considering that the words in one language could be translated into zero, one or many words in other language.

IBM model 3 introduces two more steps in translation process called the fertility and NULL insertion step which gives the solid formulation along with model 1 and 2 to build word-to-word translations. However this need further improvements to deal with large sentences where words movements are not depicted correctly.

IBM model 4 introduces the concept of relative distortion by making the words groups.

IBM model 5 resolves the problem of deficiency in model 3 and 4 where multiple words could be placed in the same position. The placement of the words is done by keeping track of vacant positions.

Homogeneous HMM is a modification of the IBM model 2 model with first-order dependencies in alignment probabilities [Vogel et al., 1996] which deals with lexicon plus relative position.

These IBM models give the comprehensive formulation to build word-to-word alignment between the sentence pairs. However, one problem persists: the asymmetric word-to-word alignment does not allow to align multiple input words to one output word. The procedure deployed to overcome this problem is to run IBM models in both directions, *i.e.* source-to-target and target-to-source. The word alignments are symmetrized by taking the intersection or union of alignment points [Och and Ney, 2003]. A comprehensive detail about the IBM models $1-5$ is described in [Och, 2002] whereas [Och and Ney, 2003] presents a systematic performance comparison of various models.

2.3.2 Phrase-based translation models

In the previous section we discussed the IBM models which use words as basic translation units. There is a reasonable point that words may not be the best option as translation unit. A word in one language may translate into many words in another language and many words in a language may translate into a single word in another one. It looks more intuitive to use group of words - called phrases - as translation units. This concept emerged as the phrase-based approach [Koehn et al., 2003]. Phrase-Based SMT (PBSMT) has proven to be one of the best performing technique. The idea behind this approach is to use a simplified version of the alignment template approach [Och and Ney, 2004]. The approach is based on phrases which are not necessarily linguistically formulated and which may be reordered as shown in figure 2.6.

Figure 2.6 illustrates many benefits of translation based on phrases instead of words. Words may not be the good atomic units for translation due to one-to-many and many-to-one mappings. For instance, the french word *sera* is translated into *will be* and an english word *month* is produced by *le mois*. By this way,

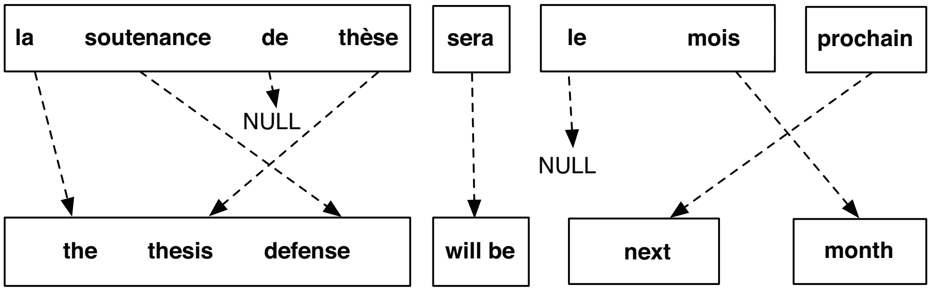

Figure 2.6: Phrase-based machine translation

translating phrases instead of single word helps to resolve translation ambiguities. Finally, the model seems much simpler and intuitive that does not allow the arbitrary adding and dropping of words. This kind of formulation urges to map phrases instead of words. PBSMT approach follows the same foundation as word-based SMT models. Equation 2.2 is further decomposed as follows:

$$P(f|e) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i) d(start_i - end_{i-1} - 1) \tag{2.8}$$

The **f** sentence in source language is decomposed into $\boldsymbol{I}$ phrases $\bar{f}_i$. The equation 2.8 constitutes two models, the first part of the product states that each source phrase $\bar{f}_i$ is translated into a target phrase $\bar{e}_i$. The term $d(start_i - end_{i-1} - 1)$ in the formula represents the *distance based reordering model*. According to this model, the reordering of a phrase is relative to the previous phrase: $start_i$ and end_i denote the start and end words of the ith source phrase that translates into ith target phrase. Since the translation direction was mathematically inverted in the noisy channel model, the phrase translation probability $\phi(\bar{f}_i|\bar{e}_i)$ is modeled from target to source.

Phrase extraction and scoring

Phrases are the core component in PBSMT. They are not supposed to be linguistically formulated. Bilingual phrases, also called *phrase pairs*, are extracted from word-to-word alignments which are contiguous and consistent. Phrases are extracted by applying a set of heuristics to the word aligned parallel corpora. According to a criterion any sequence of consecutive source words and consecutive target words which are aligned to each other and are not aligned to any other token in the sentence become a phrase. Och et al. [1999b] and Zens et al. [2002] give details of the criterion.

All words in the target language are aligned to the words in the source language and otherwise. There must be at least one word in the target language phrase which is aligned to at least one word in the source language phrase. Possible unaligned words at the boundaries of the phrases are taken into account by a phrase extraction algorithm. Formally, an alignment a having words $f_1, ..., f_n$ in $\bar{f}$ contains alignment points with words $e_1, ..., e_n$ in $\bar{e}$ creates a consistent phrase pair ($\bar{f}$, $\bar{e}$) as shown in 2.7.

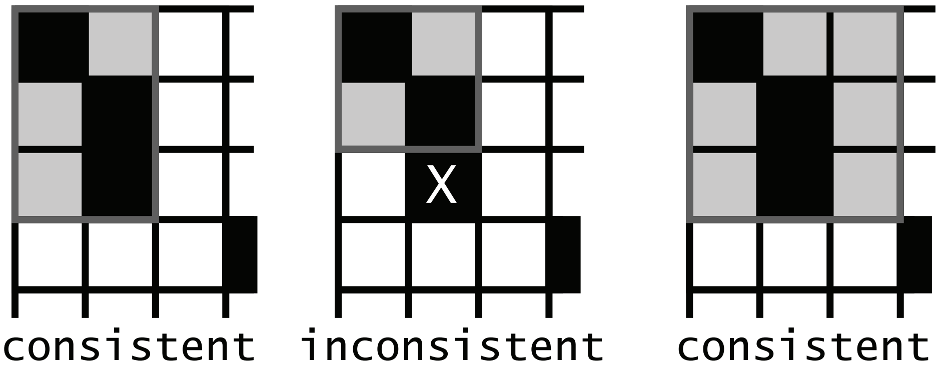

Figure 2.7: Phrase Pairs being consistent with word alignment. The grey part shows the probable phrases [Koehn, 2010].

All words have to align with each other which is the case in example 1 but violated in example 2 where one alignment point in the second column is outside the phrase pair. Example 3 includes an unaligned word which is consistent.

The phrase extraction process results into pairs of source and target phrases which have consecutive words and are consistent with the word alignment matrix. These alignments are produced in both directions since alignment is asymmetric, the intersection and/or union (or other alignment methods) of these two alignments is then used.

The phrase translation probabilities are estimated over all bilingual phrases using the relative frequency of the target sequence given the source sequence. The phrase translation probabilities $\phi(\bar{f}_i|\bar{e_i})$ are learned using the Maximum Likelihood Estimation (MLE):

$$\phi(\bar{f}_i|\bar{e_i}) = \frac{count(\bar{e}_i, \bar{f}_i)}{\sum_{\bar{f}_j} count(\bar{e}_i, \bar{f}_j)} \tag{2.9}$$

Nowadays, many SMT systems follow a phrase-based approach, in that their translation unit is the bilingual phrase, such as [Bertoldi et al., 2006; Hewavitharana et al., 2005; Matusov et al., 2006]. There are various other popular modern approaches to SMT which includes, *factored translation model* [Koehn et al., 2007a], *hierarchical approach* [Chiang, 2005, 2007; Wu, 1997], *N-gram based approach* [Casacuberta and Vidal, 2004; Casacuberta et al., 2002] and *syntax-based MT* [Och et al., 2003; Yamada and Knight, 2001].

2.4 Log-linear models

In a standard phrase-based statistical machine translation system, all the models are multiplied together, *i.e.* the translation model, the distortion model and the language model as follows:

$$e_{best} = \arg\max_{e} \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e_i}) d(start_i - end_{i-1} - 1) P_{LM}(e) \tag{2.10}$$

It seems obvious that different models may have different impact on the translation output. The available models and methods only provide poor approximations of the true probability distributions. Hence, certain models may be given more weight than others. Formally this is done by introducing the weights λ_ϕ, λ_d, λ_{LM} that scale the contribution of each of the three components. In log-linear model each of these models is considered as a feature and weighted according to the following form:

$$p(x) = exp \sum_{i=1}^{n} \lambda_i h_i(x) \tag{2.11}$$

where

$$\begin{aligned} h_1 &= \log \phi & (2.12) \\ h_2 &= \log d & (2.13) \\ h_3 &= \log P_{LM} & (2.14) \end{aligned}$$

This framework is widely used since it gives the flexibility to include many models as feature functions. The popular maximum entropy and perception learning methods are all based on log-linear models. In the next sections, we will discuss some well known models which are often used in many state-of-the-art systems.

2.4.1 Bidirectional translation probabilities

The paradigm based on Bayes rule results in the use of inverse translation direction. In practice, translation probabilities in both directions are considered in most state-of-the-art systems and it is proven that it outperforms the model based on one directional translation probabilities.

2.4.2 Lexical weighting

It is shown in Koehn et al. [2003] that the performance of the log-linear model improves by incorporating a feature which measures how well individual words translate to each other. This model is so called the lexical weighting which is computed by the product of the individual words in the phrases for the entire sentence pair as follows:

$$\mathrm{lex}(\bar{e}|\bar{f}, a) = \prod_{i=1}^{\mathrm{n}(\bar{e})} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall (i,j) \in a} w(e_i|f_j) \tag{2.15}$$

where a is the alignment between the target word positions $i = 1, ..., n$ and the source word positions $j = 1, ..., m$. $w(e_i|f_j)$ is the lexical translation probability and is estimated by relative frequency. It is also useful to use lexical translation probabilities in both directions as for phrase translation probabilities.

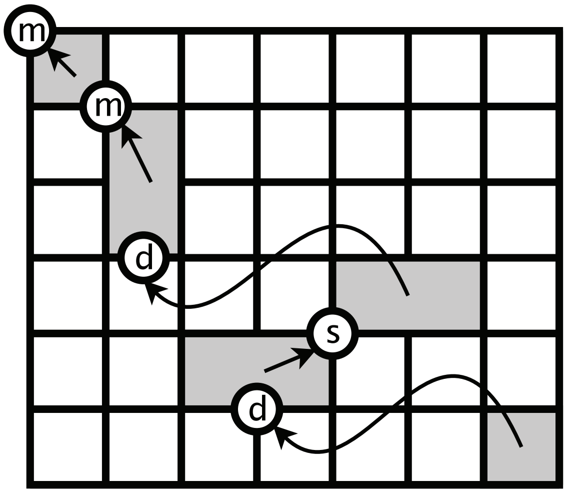

Figure 2.8: Three orientations types: (m) monotone, (s) swap, (d) discontinuous (taken from [Koehn, 2010]).

2.4.3 Lexicalized reordering model

Standard phrase-based statistical machine translation is only based on movement distance distortion model which is considered weak. It is obvious to note that some phrases are more frequently reordered than others. Therefore, lexicalized reordering model is proposed based on three orientations as shown in figure 2.8 :

- monotone : if a word alignment point to the top left exists
- swap : if a word alignment point to the top right exists
- discontinuous : neither monotone nor swap

Each extracted phrase pair is counted with each of the three orientation types and probability distribution p_o is calculated based on the maximum likelihood:

$$p_o(\text{orientation}|\bar{f},\bar{e}) = \frac{count(\text{orientation},\bar{e},\bar{f})}{\sum_o count(o,\bar{e},\bar{f})} \tag{2.16}$$

Due to the sparseness in the data to calculate the statistics of the each orientation type; the counts are smoothed with a factor σ :

$$p_o(\text{orientation}) = \frac{\sum_{\bar{f}} \sum_{\bar{e}} count(\text{orientation}, \bar{e}, \bar{f})}{\sum_o \sum_{\bar{f}} \sum_{\bar{e}} count(o, \bar{e}, \bar{f})} \tag{2.17}$$

$$p_o(\text{orientation}|\bar{f}, \bar{e}) = \frac{\sigma\ p(\text{orientation}) + count(\text{orientation}, \bar{e}, \bar{f})}{\sigma + \sum_o count(o, \bar{e}, \bar{f})} \tag{2.18}$$

There are certain variations proposed on this lexicalized reordering orientation types which are beneficial for certain language pairs.

2.4.4 Word and phrase penalty

These feature functions aim to model the output length in term of number of words and phrases. A word penalty tries to avoid too long or too short candidate sentences by introducing a feature ω for each produced word. In the case of $\omega < 1$ the score of shorter translations is increased otherwise longer translations are preferred. This parameter improves the translation performance by optimizing the length of the output.

Similarly, it is questionable whether longer or shorter phrases are better. Similar to word penalty a feature ρ is introduced for each phrase translation called phrase penalty. If $\rho < 1$ longer phrases are preferred and shorter phrases get emphasis in case of $\rho > 1$.

2.5 Language models

The language model is an important component of many natural language processing applications such as speech recognition [Chen and Goodman, 1996; Roark et al., 2007], information retrieval [Song and Croft, 1999] and SMT. It ensures the fluency of the sentence. Most SMT systems simply borrow the language models originally employed for speech recognition. According to [Lopez, 2008], the main focus of SMT research has been on translation models, but it is proven that improvements in the language model generally leads to better translation performance [Brants et al., 2007].

A statistical language model assigns a probability to a sequence of m words $P(e_1, \ldots, e_m)$ by means of a probability distribution. A simple choice for this model consists in dividing the sentences into smaller parts (n-grams), small enough to be frequent in the corpus, but large enough to contain some language context. An n-gram is a contiguous sequence of n items from a given sequence of text. When n is 1 the n-gram is referred to as a ”unigram”, when n is 2 it is called ”bigram” , when n is 3 it is a ”trigram” and so on. The probability $P(e_1, \ldots, e_m)$ of observing the sentence $e1, ..., e_m$ is approximated as:

$$P(e_1, \ldots, e_m) = \prod_{i=1}^{m} P(e_i|e_1, \ldots, e_{i-1}) \approx \prod_{i=1}^{m} P(e_i|e_{i-(n-1)}, \ldots, e_{i-1}) \qquad (2.19)$$

which means that the probability of observing the *ith* word e_i in the context history of the preceding $i-1$ words can be approximated by the probability of observing it in the shortened context history of the preceding $n-1$ words.

The conditional probability can be estimated from n-gram frequency counts as follows:

$$P(e_i|e_{i-(n-1)}, \ldots, e_{i-1}) = \frac{count(e_{i-(n-1)}, \ldots, e_{i-1}, e_i)}{count(e_{i-(n-1)}, \ldots, e_{i-1})} \qquad (2.20)$$

The derived model has major problem when observing any n-grams that has not explicitly been seen before. According to the formulation given above, n-grams not seen in the corpus will have a probability of zero and will nullify the whole sentence probability. The problem is tackled by a technique called smoothing in which a very small positive value is assigned to zero probabilities. The simplest approach is to add one to all the counts of n-grams, this is known as *add-one smoothing*, for example for bigrams:

$$P(e_j|e_i) = \frac{count(e_i e_j) + 1}{count(e_i) + V} \qquad (2.21)$$

where V is the size of the vocabulary.

The smoothing techniques include *interpolation* and *back-off* models. A good overview of $n-$gram smoothing techniques is presented in [Chen and Goodman, 1996]. Generally, back-off n-gram models are used in SMT, however various language modeling techniques exist and have shown to improve SMT quality, some of these include for instance continuous space LM based on neural networks [Schwenk, 2007; Schwenk et al., 2006], syntax based models based on context free grammars [Charniak et al., 2003; Marcu et al., 2006; Wu et al., 1998].

2.6 Decoder

So far we have introduced the two main components of an SMT system, the translation model - learned from bilingual corpus and the language model estimated from monolingual data in the target language. In order to translate an observed source sentence f, we seek the target sentence e which maximizes the product of those two terms. This process is also called decoding step. Decoding is an important part in the SMT process. Without a reliable and efficient decoding algorithm, the system could miss the best translation of an input sentence even if it is perfectly predicted by the models. The decoding process in machine translation finds the best scoring translation among many choices.

In word-based SMT systems, decoding was done with different approaches including optimal A* search [Och et al., 2001], integer programming [Germann et al., 2001], greedy search algorithms [Wang and Waibel, 1998]. An important problem of these decoders is the computational complexity introduced by reordering when single words are considered instead of longer units.

In phrase-based decoders, short-distance reorderings between source and target sentences are already captured within the translation units, which relieve the reordering problem [Koehn, 2004; Och and Ney, 2004; Tillmann and Ney, 2000]. Pharaoh [Koehn, 2004] was an efficient and freely available beam search phrase-based decoder. It was very successful and contributed in making SMT more accessible and more popular. Recently, Pharaoh has been upgraded by another decoder which is called Moses [Koehn et al., 2007a]. *Moses* is much more powerful than Pharaoh and very popular in research community. It is also a phrase-based decoder implementing a beam search, allowing to input a word lattice with confusion networks and using a factored representation of the raw words (surface forms, lemma, part-of-speech, morphology, word classes, etc.). Nowadays, many SMT systems employ a phrase-based beam search decoder because of the reasonable performance results achieved in terms of accuracy and efficiency.

2.7 Minimum error rate training

The commonly used log-linear model in SMT is a combination of several features weighted by the parameter λ_i as shown in 2.11. It is important to learn the parameter λ_i for the features h_i in order to achieve good translation performance. This can be performed by minimizing translation error over a development corpus for which manually translated references are available. This minimization in multiple dimensions is a complicated problem for given objective function. It has no analytic representation, therefore the gradient cannot be estimated. Also, it may have many local minima. Moreover, its evaluation has a significant computational cost.

Och [2003] proposed an efficient supervised algorithm -so called *Minimum error rate training* (*MERT*) to calculate the optimal weights of the parameter λ_i for the features h_i. These weights are tuned on a given development set. During *MERT* optimization, it is assumed that the best model is the one that produces the minimum overall translation error with respect to given error function. A simple pseudo code is as follows:

- Initialization : initialize λ_i randomly or heuristics based
- Translation: n-best translation of the development set with given λ_i
- Comparison: compare the objective score (such as BLEU) of the n-best translation with previous run
- Re-estimation: Re-estimate the weights λ_i
- Iterate: Iterate until weights are converged

The other popular approaches in literature are Powell's method [Powell, 1964] and the downhill simplex method [Cettolo et al., 2005; Nelder and Mead, 1965; Press et al., 2002].

2.8 Evaluation metrics

MT evaluation is a large field of research by itself. The purpose of MT evaluation is to judge the correctness of an SMT output. It can be measured according to several factors including intelligibility, fidelity, coherence, usability, adequacy and fluency. Initially, MT evaluation was performed by humans only, no automatic measures were available. Human evaluation requires human intervention in order to judge the quality of the translation. The focus of MT evaluation has been to evaluate adequacy and fluency according to a certain quality scale [White, 1994]. Fluency expresses how natural the hypothesis sounds to a native speaker of the target language while adequacy indicates how much of the information from the original translation is expressed in the translation. Many international evaluation campaigns conduct human evaluation with different grades to compare the quality of various systems. For instance, fluency is usually measured with these possible scores: 5 for *flawless*, 4 for *good*, 3 for *non-native*, 2 for *disfluent* and 1 for *incomprehensible*. To measure adequacy the judge is presented with a reference translation to grade how much of the information is present in comparison to the references. The information expressed in original translation: 5 for *all of the information*, 4 for *most of the information*, 3 for *much of the information*, 2 for *little information*, and 1 for *none of it*.

The cost of human evaluation makes it very difficult to use it in iterative system development, where regular evaluation is required to determine system performance. It can take weeks to finish and involve human labor that can not be reused. Nowadays, the focus is on doing system comparisons. Therefore, methods for automatic machine translation evaluation got attention, that is quick, inexpensive, and language-independent, that correlates highly with human evaluation, and that has little marginal cost per run. This need has resulted in emergence of various automatic evaluation metrics, however, so far the MT community has not yet accepted a unified evaluation criteria.

The automatic metrics make use of a set of test sentences for which we already have human translations, called *reference translations*. The intuition behind these metrics is that MT must be good if it resembles human translation of the same sentence [Papineni et al., 2002]. The metrics perform partial string match between the MT output and the reference translations. However, having a single reference translation may bias the evaluation towards a particular translation style, so multiple reference translations is generally preferred to take into account the diversity of translation styles. In the following, we give an overview of the most popular MT evaluation metrics:

Word Error Rate (WER) [Och et al., 1999a], also known as Levenshtein or edit distance. WER scores the sentences based on the number of insertions, deletions and substitutions required to transform the output sentence to the reference sentence. WER is considered as less appropriate for MT evaluation because a word that is translated correctly but is in the wrong location will be penalized as a deletion (in the output location) and an insertion (in the correct location). This problem motivated the use of **Position independent word Error Rate (PER)**, which regards the output and the reference sentence as unordered bags of words rather than totally ordered strings [Och et al., 1999a].

Another variant, **Translation Edit Rate (TER)** is an evaluation metric which allows block movements of words and thus takes into account the reordering of words and phrases in the translation [Snover et al., 2006]. It also measures the amount of editing that would have to be performed to change a hypothesis so that it exactly matches the reference.

An extension of TER is **Translation edit rate plus (TERp)**. It uses all the edit operations of TER along with three new edit operations: stem matches, synonym matches and phrase substitutions. Unlike TER, the TERp implementation assigns a varying cost to substitution so that a lower cost is used if the two words are synonyms, share the same stem, or are paraphrases of each other. TERp identifies words in the hypothesis and reference that share the same stem using the Porter stemming algorithm. Two words are determined to be synonyms if they share the same synonym set according to Word Net. Sequences of words in the reference are considered to be paraphrases of a sequence of words in the hypothesis if that phrase pair occurs in the TERp paraphrase phrase table. With the exception of phrase substitutions, the edit operations have fixed cost regardless of the word in question [Snover et al., 2009].

The most widely used MT evaluation metric is **BLEU**, short for *bilingual evaluation under study* [Papineni et al., 2002]. The metric works by measuring the n-gram co-occurrence between a given translation and the set of reference translations and then taking the weighted geometric mean. BLEU is a precision oriented metric as it considers the number of n-gram matches as a fraction of the total number of n-grams in the output sentence.

A variant of BLEU score is the **NIST** evaluation metric [Doddington, 2002], which also calculates how informative a particular n-gram is, the rarer a correct n-gram, the more weight it is given. The NIST score also differs in its calculation of the brevity penalty.

The **METEOR** metric [Denkowski and Lavie, 2011] was developed made to address some of the drawbacks of the BLEU metric. It is based on the weighted harmonic mean of unigram precision and unigram recall. The metric was designed after research by [Lavie and Agarwal, 2007] on the significance of recall in evaluation metrics. Their research showed that metrics based on recall consistently achieved higher correlation to human evaluation than those based on precision alone, such as BLEU and NIST. METEOR also includes other features, such as synonym matching, where instead of matching only the exact word form, the metric also matches on synonyms. For example, the word ”nice” in the reference rendering as ”beautiful” in the translation counts as a match. The metric also includes a stemmer, which lemmatizes words and matches on the lemmatized forms. The implementation of the metric is modular insofar as the algorithms that match words are implemented as modules, and new modules that implement different matching strategies may easily be added.

2.9 Adaptation techniques

Statistical machine translation systems are learned with bilingual data. These systems are evaluated on a test corpus that is distinct from the training data. In most cases, the test corpus belongs to the same subject as the original training corpus and therefore may belong to the same domain - so called *in-domain*. To evaluate SMT quality in a different domain than the one for which the system was trained, a corpus from the different domain is required. *Out-of-domain* describes the type of training corpus, which is not subject of the current test. The corpora could be distinguished based on many factors. For instance, spoken text versus written text, formal versus informal style., human translated vs automatically extracted. There is large variety of possible domains, some of which are news, political speech, talk shows, scientific press, teaching, preaching, sport, interviews, law, political debate and business. Even within a certain domain there could be many sub-domains, for instance, scientific press may have different scientific subjects ranging from biological sciences to computer technology.

The purpose of SMT adaptation is to improve translation performance on a specific domain. This is particular important for SMT systems since their performance heavily depends on the training data domain. Data can be very different in many aspects such as size, vocabulary, style, quality or genre. Since the statistical methods are heavily influenced by both domain differences and noise, model adaptation is one of active research area in statistical machine translation. There have been several attempts in recent years that outline various techniques to adapt the models to the domain of interest. Model adaptation usually concentrated around language model or translation model adaptation.

[Béchet et al., 2004] enumerated various approaches to language model adaptation as follows:

- to train an LM in the new domain if sufficient data are available.
- pool data of various domains with the data from the new domain.
- linearly interpolate the general and a domain-specific model as shown in [Seymore and Rosenfeld, 1997].
- back-off domain specific probabilities with those of the general model.
- retrieve documents relevant to the new domain and training a new LM on-line with those data [Iyer and Ostendorf, 1996].
- apply maximum entropy and minimum discrimination adaptation [Chen et al., 1998].
- adapt with maximum posteriori Probability (MAP) [Seymore and Rosenfeld, 1997].

- adapt by linear transformation of vectors of bigram counts in a reduced space.

Generally, these approaches are being studied in the context of speech processing, but some of them are successfully applied to SMT. For instance, Zhao et al. [2004a] studied the linear-interpolation of out-of-domain and in-domain models. They retrieved documents from a large monolingual text which are similar to the desired domain and they build a new in-domain LM. This in-domain LM is linearly interpolated with the generic out-of-domain LM.

[Wu et al., 2008] experimented with two approaches for language model interpolation. They linearly interpolate LMs in a first approach while in another one they considered each language model as distinct feature in a log-linear paradigm. They showed that linear interpolation performs better.

Domain adaptation seems to be more tricky for the translation model. Today current best practice to build an SMT system is to concatenate all available parallel data, to perform word alignment and to extract and score the phrase pairs by simple relative frequency. Doing this, the parallel data is (wrongly) considered as one homogeneous pool of knowledge. To the best of our knowledge, there is no commonly accepted method to weight the bitexts coming from different sources so that the translation model is best optimized to the domain of the task.

In previous work, translation model adaptation is done by using mixture models, by self-enhancement of translation models, by exploiting comparable corpora, by data selection and by data weighting. We will summarize these approaches in the following sections.

2.9.1 Mixture models

Mixture models correspond to the mixture distribution of data to estimate probability. Assume that a corpus is composed of N different domains from a total of size S, where each domain corresponds to one of K possible topics. The distribution of such corpus could be modeled as a mixture of K different S-dimensional distribution. These kind of models are termed as topic models. In context of SMT, the mixture models could be applied at different levels. For instance, during word-to-word alignment to extract topic-dependent alignments, to construct specific language models, to adapt translation models by mixture components etc.

Civera and Juan [2007] proposed a model that can be used to generate topic-dependent alignments by extension of the HMM alignment model and derivation of Viterbi alignments. Foster and Kuhn [2007] applied a mixture model approach to adapt the system to a new domain by using weights that depend on text distances to mixture components. The training corpus was divided into different components, a model was trained on each part and then weighted appropriately for the given context. Koehn and Schroeder [2007] used two language models and two translation models: one in-domain and other out-of-domain to adapt the system. Two decoding paths were used to translate the text.

2.9.2 Self-enhancing approach

Another direction of research is self-enhancing of the translation model. This was first proposed by Ueffing [2006]. The idea is to translate the test data, to filter the translations with the help of a confidence score and to use the most reliable ones to train an additional small phrase table that is jointly used with the generic phrase table. This could be also seen as a mixture model with the in-domain component being build on-the-fly for each test set. In practice, such an approach is probably only feasible when large amounts of test data are collected and processed at once, *e.g.* a typical evaluation set up with a test set of about 50k words. This method of self-enhancing the translation model seems to be more difficult to apply for on-line SMT, e.g. a WEB service, since often the translation of some sentences only is requested. Ueffing [2007] further refined this approach by using transductive semi-supervised methods for effective use of monolingual data from the source text. A related approach was investigated in [Schwenk, 2008; Schwenk and Senellart, 2009] in which lightly supervised training was used. An SMT system was used to translate large collections of monolingual texts, which were then filtered and added to the training data. Although this technique seems to be close to self enhancing as proposed in [Ueffing, 2006] there is a conceptual difference. They do not use the test data to adapt the translation model, but large amounts of monolingual training data in the source language and they create a complete new model that can be applied to any test data without additional modification of the system. This kind of adapted system can be used in WEB service. In an extended approach, Lambert et al. [2011] argued that the automatic translations should not be performed from the source to the target language, but in the opposite direction. Secondly, they proposed to use the segmentation obtained during translation instead of performing word alignments. Finally, they proposed to enrich the vocabulary of the adapted system by detecting untranslated words and automatically inferring possible translations from the stemmed form and the existing translations in the phrase table. Bertoldi and Federico [2009] exploited large in-domain monolingual corpora either in source or in target language. In an another approach Chen et al. [2008] performed domain adaptation simultaneously for the translation, language and reordering model .

2.9.3 Comparable corpora

Because of the fact that parallel data in various specific domains is a sparse resource, it is an alternate choice to explore non-parallel data which could provide additional information somehow.

Comparable corpora is of set of texts in different languages that are not translations of each other. It is a collection of texts composed independently in the respective languages and combined on the basis of similarity of content. These are documents in one to many languages, that are comparable in content and form in various degrees and dimensions. The good amount of availability of these comparable corpora and the potential for parallel corpus as well as dictionary creation has motivated an interest in trying to make maximum use of these comparable data.

In SMT the comparable corpora are exploited to find additional parallel texts. Information retrieval techniques are used to identify candidate sentences Hildebrand et al. [2005]. Snover et al. [2008a] used cross-lingual information retrieval to find texts in the target language that are related to the domain of the source texts. Comparable corpora has been used for language and translation model adaptation in [Snover et al., 2008b]. Munteanu and Marcu [2005] proposed a technique to improve SMT performance using extracted parallel sentences. Candidate sentences are determined based on word overlap and the decision whether a sentence pair is parallel or not is performed by a maximum entropy classifier trained on parallel sentences. In a similar technique, Abdul-Rauf and Schwenk [2009] bypass the need of the bilingual dictionary by using proper SMT translations and instead of a maximum entropy classifier they used simple measures like the word error rate (WER) and the translation error rate (TER) to decide whether sentences are parallel or not.

2.9.4 Data selection

Another line of research in domain adaptation is data selection. Data selection is the process of determining and selecting the appropriate data to a given task among all available data. The approach of data selection has been studied for both monolingual data to adapt the language models as well as for bilingual data to adapt the translation models. Most of the techniques rely on information retrieval, perplexity or cross entropy. Zhao et al. [2004a] constructed specific language models by using machine translation output as queries to extract similar sentences from large monolingual corpora. Hildebrand et al. [2005] assumed that the general corpus is composed of different domain sub-corpora. Therefore, they filtered the large bilingual out-of-domain corpus to select those sentence pairs only, which belongs to the in-domain test set. By these means, the bilingual out-of-domain corpus is filtered to a bilingual in-domain corpus. In an another work two approaches are explored in Lu et al. [2007]. They proposed offline versus on-line optimization. Offline data optimization is done by using information retrieval to select similar data to a given task and redistribute weight to each sentence. In on-line optimization process, several translation models and a general one are created. During the translation, similarity between the input and translation models is calculated and given certain weight. Another approach for translation model adaptation is investigated in Axelrod et al. [2011] by using cross-entropy to select so called pseudo in-domain sentences from general domain corpus. They proposed to rank the sentences in a general-domain corpus with respect to an in-domain corpus. A cutoff can then be applied to produce a very small sub-corpus, which in turn can be used to train a domain-adapted MT system. They showed that it is possible to use data selection methods to sub-select less than 1% of a large general training corpus and still increase translation performance.

2.9.5 Data weighting

Most recently, weighting the data is getting much attention from the research community. Various features extracted at different levels during model training are considered to weight the data. The data with a higher feature scores is given higher weights.

Matsoukas et al. [2009] proposed a technique in which they weighted each sentence in the training bitexts to optimize a discriminative function on a given tuning set. Sentence level features were extracted to estimate the weights that are relevant to the given task. The feature vectors were mapped to scalar weights $(0, 1)$ which are then used to estimate probabilities with weighted counts. Their technique is only concerned with influencing the translation probabilities via the corpus weights; it does not change the set of rules extracted.

Foster et al. [2010] proposed an extended approach by an instant weighting scheme which learns weights on individual phrase pairs instead of sentences and incorporated the instance-weighting model into a linear combination. Phillips and Brown [2011] trained the models with a second-order Taylor approximation of weighted translation instances and discount models on the basis of this approximation. Zhao et al. [2004b] rescore phrase translation pairs for statistical machine translation using TF.IDF to encode the weights in phrase translation pairs. The translation probability is then modeled by similarity functions defined in a vector space. Huang and Xiang [2010] proposed a rescoring algorithm in which phrase pair features are combined with linear regression model and neural network to predict the quality score of the phrase translation pair. These phrase scores are used to boost good phrase translations and bad translations are discarded. Sennrich [2012] demonstrated perplexity optimization for weighted counts. They also show that variable features in translation table could be optimized separately through perplexity optimization.

Among other adaptation techniques Duh et al. [2011]; Sanchis-Trilles and Casacuberta [2010] proposed Bayesian adaptation, cache-based adaptation Nepveu et al. [2004].

2.10 Conclusion

In this chapter, we have presented various MT concepts. After the brief description of different MT approaches , we have focused on various SMT paradigms from word-to-word alignment to phrase based approaches. The concepts of translation and language modeling along with decoding and evaluation metrics are explained in detail. The various techniques for model adaptation in literature are categorized according to underlying methodologies.

The work proposed in this thesis is an extension and generalization of several ideas proposed in previous works such as weighted counts with feature scores. However our proposed framework gives the flexibility to inject the feature scores in a unified formulation calculated at various levels. It is based on the following principles:

- the use of a set of “quality measures” at different levels: weights for each corpus (or data source) and for each individual sentence in the bitexts.

- There are various methods to estimate the probability distribution of the models given the training corpora, and it may be difficult to integrate weights at the corpus or sentence level directly in this procedure. Therefore we started by weighting corpora without assuming how translation probabilities are estimated. The idea we proposed is to use resampling to produce a new collection of weighted alignment files. A weighting coefficient to each bitext is associated. This method does not require an explicit specification of the in-domain and out-of-domain training data. The weights of the corpora are directly optimized on the development data using a numerical method, similar to the techniques used in the standard minimum error training of the weights of the feature functions in the log-linear criterion.

- The second idea proposed in this thesis is to consider some kind of meta-weights for each part of the training data. Instead of numerically optimizing all the weights, these meta-weights only depend on few parameters that need to be optimized. The weighting of the parts is still done by resampling the alignments.

- In a third approach , we integrate the various weighting schemes directly in the calculation of the translation probabilities. Resampling the bitexts or alignments is computationally expensive for large corpora since the resampled data is ideally much bigger than the original one. Instead, we worked on direct translation probabilities.

- our approach has only a small number of parameter to optimize.

- No additional feature functions to express the quality or appropriateness of certain phrase pairs, but we modify only the existing phrase probabilities. By these means, we don't have to deal with the additional complexity of decoding and optimizing the weights of many feature functions.

The novel approaches are explained in next chapters along with their detailed architecture and experimental evaluation.

Chapter 3

Weighting Data by Resampling

3.1 Background

Recently weighting the data coming from different sources belonging to different domains has attained a great attention by research community in machine translation. This is because of the fact that there is still a huge potential to get maximum out of all available sparse resources specially the parallel texts, which is available in limited domains. Considering the fact, that the performance of an SMT system is proportional to the quantity of training data used to build these systems, data is collected and merged together regardless of its nature/domain, size and quality. This introduces the sample bias problem. In practice, all the text available is biased in one or other way. For example, the UN proceedings mostly contain the political discussions and data is biased towards political domain, the text collected from news agencies is centered around news domain. Any statistics calculated from the biased sample are erroneous and can lead to under or over representation of related parameters. In other words, it will not accurately represent the target domain.

3.2 Overview of proposed schemes

The translation model of a statistical machine translation systems is trained on parallel data coming from various sources and domains.

In the following, we will first summarize how the phrase-table is calculated in the popular Moses SMT toolkit. Each research team has its own heuristics, but we assume that the basic procedure is very similar for most phrase-based systems. Moses uses four probabilities: the forward phrase-translation probability $P(\tilde{e}|\tilde{f})$, the backward phrase-translation probability $P(\tilde{f}|\tilde{e})$, and two lexical probabilities, again in the forward and backward direction. These probabilities are used in the standard log-linear model as feature functions $ft_i(f,e)$:

$$e^* \quad = \quad \arg\max_e \sum_i \lambda_i \log ft_i(f,e) \qquad (3.1)$$

Moses uses in total fourteen feature functions: the above mentioned four scores for the phrases, a phrase and word penalty, six scores for the lexicalized distortion model, a language model score and a distance based reordering model.

The phrase-table itself is created by the following procedure:

1. collect parallel training data
2. eventually discard sentence pairs that are too long or which have a large length difference
3. run Giza++ on this data in both directions (source-to-target and target-to-source)
4. use some heuristics to symmetrize the alignments in both directions, *e.g* the so-called *grow-diagonal-final-and* Koehn et al. [2003]
5. extract a list of phrases
6. calculate the lexical probabilities
7. calculate the phrase probabilities $P(\tilde{e}|\tilde{f})$ and $P(\tilde{f}|\tilde{e})$.
8. create the phrase table by merging the forward and backward probabilities

During all these steps, the corpora are not weighted according to their importance to the domain of the translation task. This is in contrast to the training

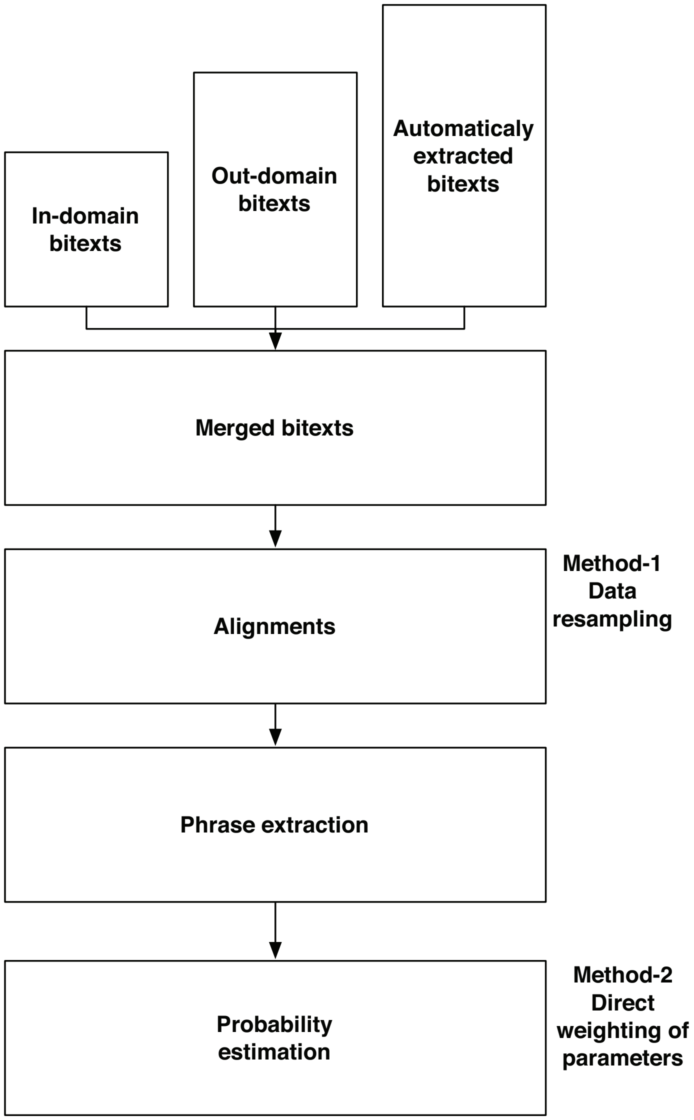

Figure 3.1: Proposed weighting schemes at different steps during translation model creation.

of the language model for which well known techniques are used to weight the various sources of texts. We propose two methods to adapt the translation model of an SMT system as shown in figure 3.1.

In a first method, the sample bias problem is adjusted by picking a random sample, which result into a better approximations of the related parameters. The main idea is to use resampling to produce a new collection of alignment files, followed by the standard procedure to extract the phrases. In a second step, we also consider the alignment score of each parallel sentence pair. By these means, the good alignments are emphasized and the less reliable ones are down-weighted.

All the alignments of the bitexts are resampled and given equal chance to be selected and therefore influence the translation model in a different way. Our proposed technique does not require the calculation of extra sentence level features, however, it may use the alignment score associated with each sentence pair as a confidence score.

In a second approach, we have employed direct weighting of phrase translation probabilities. It is done by weighted counts of phrase pairs extracted from the parallel data. Further, some features are considered for each phrase pair, which directly effects the probability estimations and results into better system performance. We will discuss this approach in detail in chapter 5.

We only perform experiments with phrase-based systems, but the methods are generic and could be easily applied to a hierarchical system. All the parameters of our procedure are automatically tuned by optimizing the BLEU score on the development data. Our method does not require an explicit specification of the in-domain and out-of-domain training data. The weights of the corpora are directly optimized on the development data using a numerical method, similarly to the techniques used in the standard minimum error training of the weights of the feature functions in the log-linear criterion.

The rest of the chapter is organized as follows. Section 3.3 describes the architecture allowing to resample and to weight the bitexts. Experimental results are presented in section 3.4 and the chapter concludes with a discussion.

3.3 Description of the resampling algorithm

The architecture of the algorithm is summarized in figure 3.2. The starting point is an (arbitrary) number of parallel corpora. We first concatenate these bitexts and perform word alignments in both directions using GIZA++. This is done on the concatenated bitexts since GIZA++ may perform badly if some of the individual bitexts are rather small. Next, the alignments are separated in parts corresponding to the individual bitexts and a weighting coefficient is associated

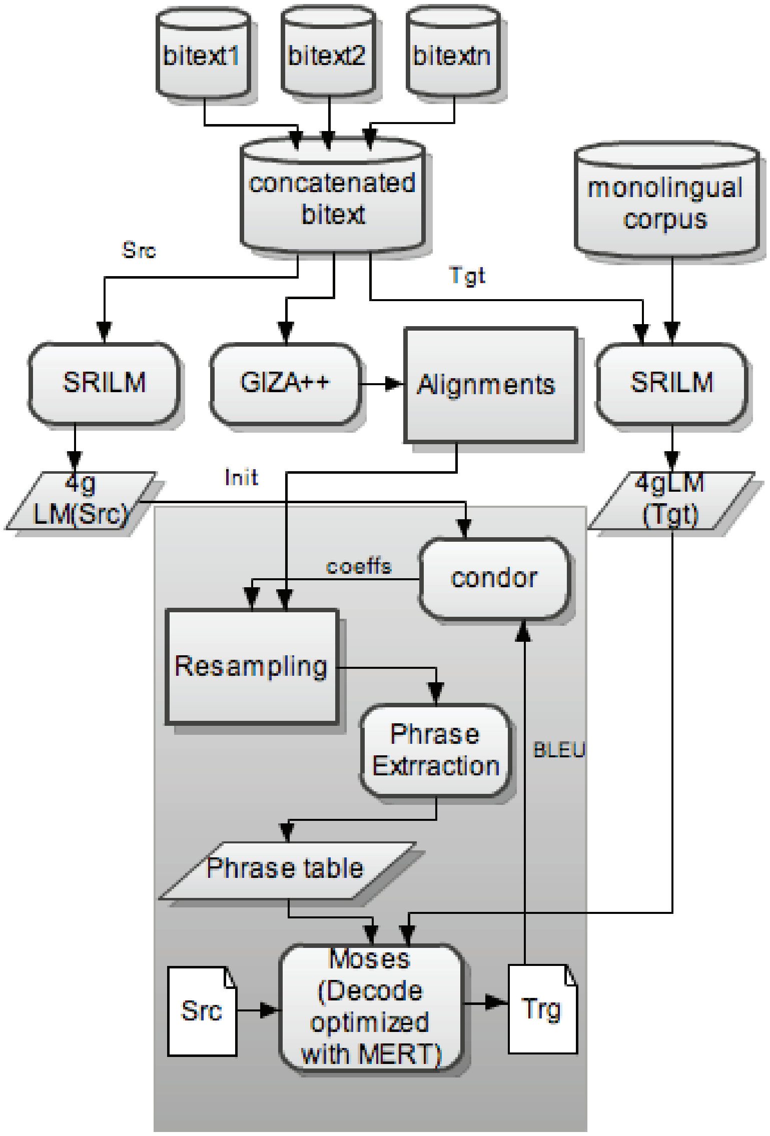

Figure 3.2: Architecture of SMT Weighting System

to each one. We are not aware of any procedure to calculate these coefficients in an easy and fast way without building an actual SMT system. Note that an EM procedure exists to do this for language modeling.

In the next section, we will experimentally compare equal coefficients, coefficients set to the same values than those obtained when building an interpolated language model on the source language, and a new method to determine the coefficients by optimizing the BLEU score on the development data.

One could imagine to directly use these coefficients when calculating the various probabilities of the extracted phrases which will be presented in chapter 5.

In this chapter, we propose a different procedure that makes no assumptions on how the phrases are extracted and probabilities are calculated. The idea is to *resample the alignments of each corpus* according to the weighting coefficient associated to the corpus. By these means, we create a new, potentially larger alignment file, which will in turn be used by the standard phrase extraction procedure.

3.3.1 Resampling the alignments

In statistics, resampling is based upon repeated sampling within the same sample until a sample is obtained which better represents a given data set Yu [2003]. Resampling is used for validating models on given data set by using random subsets. It overcomes the limitation about making assumptions about the distribution of the data. The more often we resample, the closer we get to the true probability distribution.

In our case we performed resampling with replacement according to the following algorithm:

Algorithm 1 *Resampling*

```
for $i = 0$ to required size do
    Select any alignment randomly
    $Al_{score} \leftarrow$ normalized alignment score
    $Threshold \leftarrow$ rand[0, 1]
    if $Al_{score} > Threshold$ then
        keep it
    end if
end for
```

Let us call resampling factor, the number of times resampling is applied. An interesting question is to determine the optimal value of this resampling factor.

It actually depends upon the task or data we are experimenting on. We may start with one time resampling and could stop when results become stable. Figure 3.3 shows the BLEU score as a function of the number of times we resample. It can be observed that the curve is growing proportionally to the resampling factor until it becomes stable after a certain point.

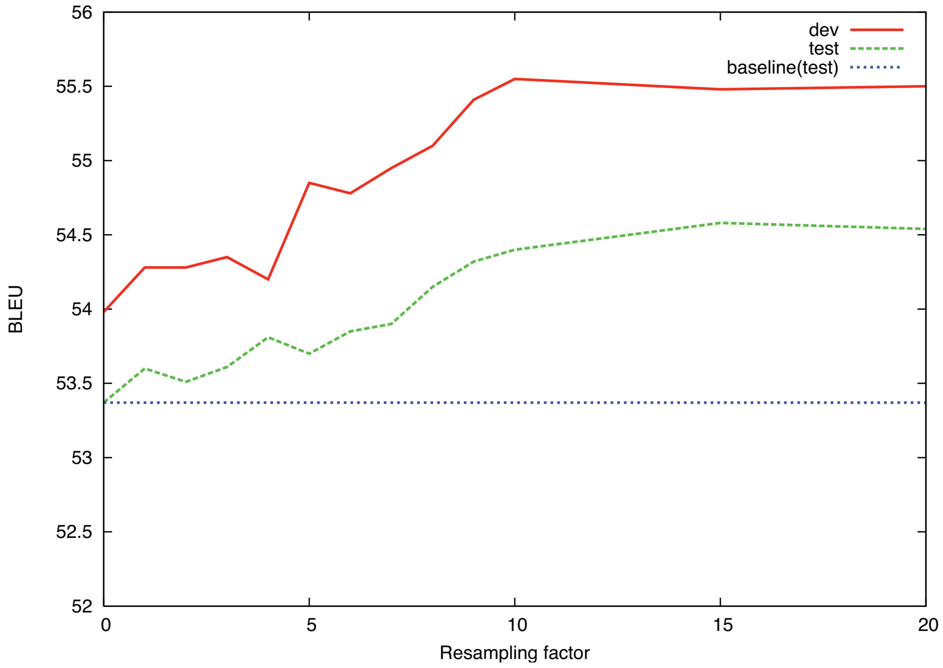

Figure 3.3: The curve shows that by increasing the resampling factor we get better and stable results on Dev and Test.(on IWSLT'09 task - section 3.4)

3.3.2 Weighting Schemes

We concentrated on translation model adaptation when the bitexts are heterogeneous, *e.g.* in-domain and out-of-domain or of different sizes. In this case, weighting these bitexts seems interesting and can be used in order to select data which better represents the target domain. Secondly, some sentence pairs are more reliable or useful than others. Using unreliable alignments can put a negative effect on the translation quality. So, we need to exclude or down-weight unreliable alignments. Weighting data can be done at two different steps that are:

- weighting the corpora: the goal is to give more importance to in-domain data

- weighting the alignments: the goal is to discard or down-weight unreliable sentence pairs

Weighting Corpora

We started to resample the bitexts with equal weights to see the effect of resampling. This gives equal importance to each bitext without taking into account the domain of the text to be translated. However, it should be better to give appropriate weights according to a given domain as shown in equation 3.2

$$\alpha_1 bitext_1 + \alpha_2 bitext_2 + .. + \alpha_n bitext_n \tag{3.2}$$

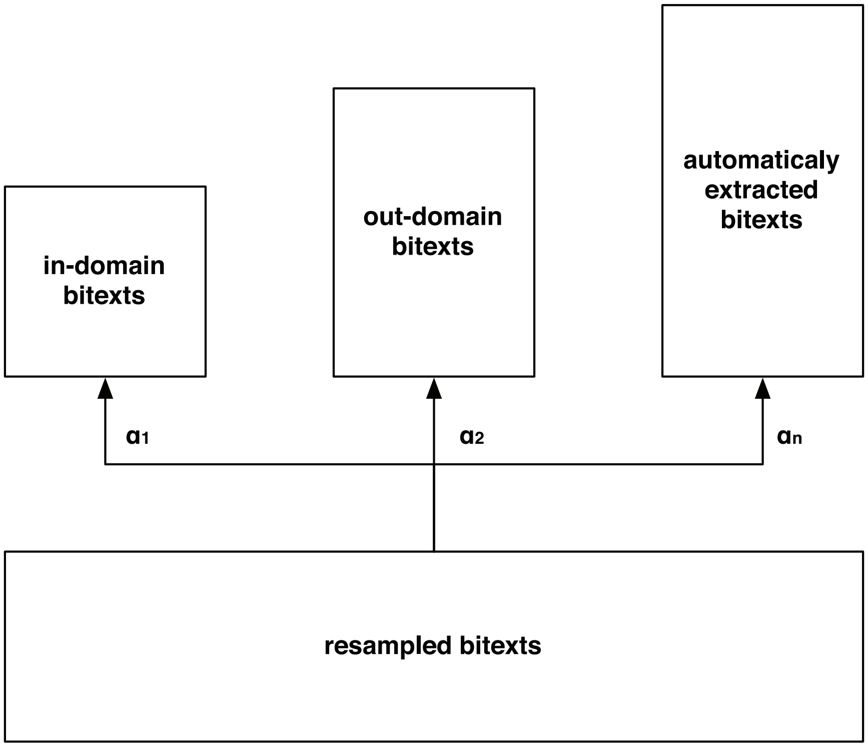

Figure 3.4: Proportion of data selected from each bitexts based on coefficients α_n

where the α_n are the coefficients to optimize representing the proportion of the data to be selected from corpus n as shown in figure 3.4 . One important question is how to find out the appropriate coefficient for each corpus.

There exists a well known technique to estimate interpolation weights in order to merge various language models into one improved LM. Those weights are estimated by an EM procedure which tends to minimize the perplexity (computed on an in-domain development corpus) of the resulting LM. They also can be seen as coefficients representing the importance of each individual model and therefore the importance of the data from which the model has been learned.

We have considered to use those coefficients to weights the available bitexts. To do so, we built an interpolated LM on the source side of the corpus and used the estimated coefficients as corpus weights. One can certainly ask the question whether the perplexity is a good criterion for weighting bitexts.

Therefore, we worked on direct optimization of these coefficients by CONDOR [Berghen and Bersini, 2005]. This freely available tool is a numerical optimizer based on Powell's UOBYQA algorithm Powell [1994]. The aim of CONDOR is to minimize an objective function using the least number of function evaluations. Formally, it is used to find $x^* \in R^n$ with given constraints which satisfies

$$F(x^*) = \min_x F(x) \tag{3.3}$$

where n is the dimension of search space and x^* is the optimum of x. The following algorithm was used to weight the bitexts.

Algorithm 2 $WeightingCorpora$

```
Determine word to word alignment with GIZA++ on concatenated bitext
Initialize CONDOR with LM interpolation weights
while Not converged do
    Create new alignment file by resampling according to weights given by CONDOR
    Use the alignment file to extract phrases and build the translation table (phrase table)
    Tune the system with MERT
    Calculate the BLEU score on the development corpus
end while
```

During experiments, the tuning step can be skipped until weights are optimized in order to save time. This has not been shown to produce worse results.

Weighting Alignments

The alignments produced by GIZA++ have an alignment score associated with each sentence pair in both direction, *i.e.* source to target and target to source. These alignment scores are meant to reflect the quality of the alignment produced. In order to refine the resampling, we used these alignment scores as a confidence measurement during the selection process. However, they can't be used as is because they depend upon the length of each sentence, and are therefore not comparable one with each other. Consequently, they have to be normalized in function of the sentence length.

Also, alignment scores have a very large dynamic range with a sharp distribution which make them hardly discriminant. The following logarithmic mapping has been used in order to flatten the distribution of the scores:

$$\log(\lambda \cdot \frac{(\sqrt[n_{trg}]{a_{src\_trg}} + \sqrt[n_{src}]{a_{trg\_src}})}{2}) \tag{3.4}$$

where a is the alignment score, n is the size of a sentence and λ is a coefficient to optimize. This is also done by CONDOR.

Of course, some alignments will appear several times, but this will increase the probability of certain phrase-pairs which are supposed to be more related to the target domain. We have observed that the weights of an interpolated LM built on the source side of the bitext are good initial values for CONDOR. Moreover, weights optimized by CONDOR are in the same order than these "LM weights". Therefore, we do not perform MERT of the SMT systems build at each step of the optimization of the weights α_i and λ by CONDOR, but use the values obtained by running MERT on a system obtained by using the "LM weights" to weight the alignments. Once CONDOR has converged to optimal weights, we can then tune our system by MERT. This saves lot of time taken by the tuning process and it had no impact on the results.

3.4 Experimental evaluation

We considered Arabic to English translation. The baseline system is a standard phrase-based SMT system based on the Moses toolkit Koehn et al. [2007b]. In

	IWSLT Task		NIST Task	
	Dev (Dev6)	Test (Dev7)	Dev (NIST06)	Test (NIST08)
Baseline	53.98	53.37	43.16	42.21
With equal weights	53.71	53.20	43.10	42.11
With LM weights	54.20	53.71	43.42	42.22
CONDOR weights	54.80	53.98	43.49	42.28

Table 3.1: BLEU scores when weighting corpora (one time resampling)

	IWSLT Task		NIST Task	
	Dev (Dev6)	Test (Dev7)	Dev (NIST06)	Test (NIST08)
Baseline	53.98	53.37	43.16	42.21
With equal weights	53.80	53.30	43.13	42.15
With LM weights	54.32	53.91	43.54	42.37
CONDOR weights	55.10	54.13	43.80	42.40

Table 3.2: BLEU scores when weighting corpora (optimum number of resampling)

our system we used fourteen features functions. These features functions include phrase and lexical translation probabilities in both directions, seven features for lexicalized distortion model, a word and phrase penalty, and a target language model. The MERT tool is used to tune the coefficients of these feature functions. The tokenization of the Arabic source texts is done by a tool provided by SYSTRAN which also performs a morphological decomposition. We considered two well known official evaluation tasks to evaluate our approach, namely NIST OpenMT and IWSLT.

For IWSLT evaluation, we used the BTEC bitexts (194K words), Dev1, Dev2, Dev3 (60K words each) as training data, Dev6 as development set and Dev7 as test set. From previous experiments, we have evidence that the various development corpora are not equally important and weighting them correctly should improve the SMT system. We analyzed the translation quality as measured by the BLEU score for the three conditions: equal weights, LM weights and optimized weights. In all cases, the resampling has been done only once. Further experiments were performed using the optimized number of resampling with and

	IWSLT Task		
	Dev (Dev6)	Test (Dev7)	TER(Test)
Baseline	53.98	53.37	32.75
With equal weights	53.85	53.33	32.80
With LM weights	54.80	54.10	31.50
CONDOR weights	55.48	54.58	31.31

	NIST Task		
	Dev (NIST06)	Test (NIST08)	TER(Test)
Baseline	43.16	42.21	51.69
With equal weights	43.28	42.21	51.72
With LM weights	43.42	42.41	51.50
CONDOR weights	43.95	42.54	51.35

Table 3.3: BLEU and TER scores when weighting corpora and alignments (optimum number of resampling)

without weighting the alignments. We have realized that it is beneficial to always include the original alignments. Even if we resample many times there is a chance that some alignments might never be selected but we do not want to loose any information. By keeping original alignments, all alignments are given a chance to be selected at least once. All these results are summarized in tables 3.1, 3.2 and 3.3.

One time resampling along with equal weights gave worse results than the baseline system while improvements in the BLEU score were observed with LM and CONDOR weights for the IWSLT task, as shown in table 3.1. Resampling many times always gave more stable results, as already shown in figure 3.3 and as theoretically expected. For this task, we resampled 15 times. The improvements in the BLEU score are shown in table 3.2. Furthermore, using the alignment scores resulted in additional improvements in the BLEU score. For the IWSLT task, we achieved an overall improvement of 1.5 BLEU points on the development set and 1.2 BLEU points on the test set as shown in table 3.3

IWSLT Task	BTEC	Dev1	Dev2	Dev3
# of Words	194K	60K	60K	60K
LM Coeffs	0.7233	0.1030	0.0743	0.0994
Optimized Coeffs	0.6572	0.1058	0.1118	0.1253

NIST TASK	Gale	NewsWire	TreeBank	Dev	ISI
# of words	1.6M	8.1M	0.4M	1.7M	43.7M
LM Coeffs	0.3215	0.1634	0.0323	0.1102	0.3726
Optimized Coeffs	0.4278	0.1053	0.0489	0.1763	0.2417

Table 3.4: Weights of the different bitexts.

To validate our approach, we further experimented with the NIST OpenMT evaluation task. Most of the training data used in our experiments for the NIST OpenMT task is made available through the LDC. The bitexts consist of texts from the GALE project[1] (1.6M words), various news wire translations[2] (8.0M words) on development data from previous years (1.6M words), LDC treebank data (0.4M words) and the ISI extracted bitexts (43.7M words). The official NIST06 evaluation data was used as development set and the NIST08 evaluation data was used as test set. The same procedure was adapted for the NIST OpenMT task as for the IWSLT task. Results are shown in table 3.1 by using different weights and one time resampling. Further improvements in the results are shown in table 3.2 with the optimum number of resampling which is 10 for this task. Finally, results by weighting alignments along with weighting corpora are shown in table 3.3. Our final system achieved an improvement of 0.79 BLEU points on the development set and 0.33 BLEU points on the test set. TER scores are also shown on test set of our final system in table 3.3. Note that these results are state-of-the-art when compared to the official results of the 2008 NIST OpenMT evaluation[3].

The weights of the different corpora are shown in table 3.4 for the IWSLT and NIST OpenMT task. In both cases, the weights optimized by CONDOR are substantially different form those obtained when creating an interpolated

[1] LDC2005E83, 2006E24, E34, E85 and E92

[2] LDC2003T07, 2004E72, T17, T18, 2005E46 and 2006E25.

[3] http://www.nist.gov/speech/tests/mt/2008/

LM on the source side of the bitexts. In any case, the weights are clearly non uniform, showing that our algorithm has focused on in-domain data. This can be nicely seen for the NIST OpenMT task. The Gale texts were explicitly created to contain in-domain news wire and WEB texts and actually get a high weight despite their small size, in comparison to the more general news wire collection from LDC.

3.5 Conclusion

We have proposed a new technique to adapt the translation model by resampling the alignments, giving a weight to each corpus and using the alignment score as confidence measurement of each aligned phrase pair. Our technique does not change the phrase pairs that are extracted,[1] but only the corresponding probability distributions. By these means, we hope to adapt the translation model in order to increase the weight of translations that are important to the task, and to down-weight the phrase pairs which result from unreliable alignments.

We experimentally verified the new method on the low-resource IWSLT and the resource-rich NIST'08 OpenMT tasks. We observed significant improvement on both tasks over the state-of-the-art baseline systems. This weighting scheme is generic and it can be applied to any language pair and target domain. We made no assumptions on how the phrases are extracted and it should be possible to apply the same technique to other SMT systems which rely on word-to-word alignments.

On the other hand, our method is computationally expensive since the optimization of the coefficients requires the creation of a new phrase table and the evaluation of the resulting system in the tuning loop. Note however, that we run GIZA++ only once.

Finally, it is straight forward to consider more feature functions when resampling the alignments. This may be a way to integrate linguistic knowledge into the SMT system, *e.g.* giving low scores to word alignments that are *"grammatically not reasonable"*.

[1] when also including the original alignments

Chapter 4

Parametric Weighting of Data

As discussed in chapter 1, the parallel data is inhomogeneous with respect to various factors. One of them is the recency of the data with respect to the given task. In principle, data with less temporal distance will most likely be similar with respect to style and vocabulary. This so-called *recency effect* is of interest specially in the news domain where named entities, etc change over time. We could for instance consider the translation of a continuous stream of news. Adapting the system to recent texts should be beneficial.

[Hardt and Elming, 2010] have shown a recency effect in terms of file-context and concluded that data within the same file is of greater importance than the rest. In another work [Levenberg et al., 2010] proposed an incremental training procedure to deal with a continuous stream of parallel text. Word alignment was performed by the stepwise online EM algorithm and the phrase table was represented with suffix arrays. The authors showed that it is better to use parallel data close to the test data than all the available data.

4.1 Overview of the idea

The work presented in this chapter is an extension of the ideas discussed in the previous chapter to weight corpora by resampling and the work of Levenberg et al. [2010] to consider the recency of the training data. In fact, we could split the training data into several parts over the time scale and use the resampling approach discussed in the previous chapter to automatically optimize the weights of each time period. However, this approach does not scale very well when the number of individual corpora increases. Numerical optimization of more than ten corpus weights would probably need a large number of iterations, each one consisting in the creation of a complete phrase table and its evaluation on the development data.

Therefore, the main idea is to consider some kind of meta-weights for each part of the training data. Instead of numerically optimizing all the weights, these meta-weights only depend on few parameters that need to be optimized. Concretely, in this work we study the exponential decrease of the importance of parallel data in function of its temporal distance to the development and test data. The weighting of the parts is still done by resampling the alignments. However, our general approach is not limited to weighting the training data with respect to recency to the development and test data. Any other criterion could be used as long as it can be calculated by a parametric function, *e.g.* to measure the topic appropriateness.

The rest of chapter is organized as follows. Section 4.2 describes the architecture of the weighting scheme. Section 4.3 explains the algorithm. Experimental results are presented in section 4.4 and the chapter concludes with a discussion.

4.2 Architecture of weighting scheme

The main idea of our work is summarized in Figure 4.1. We consider that time information is available for the bitexts. If this is not the case, one can consider that the time advances sequentially with the lines in the file. First, the data is considered in parts according to the time information. In Figure 4.1, we group together all data within the same year, but any other granularity is possible (months, weeks, days, etc). Given the observation that more recent training data seems to be more important than older one, we apply an exponential decay function:

$$e^{-\lambda \cdot \Delta t} \tag{4.1}$$

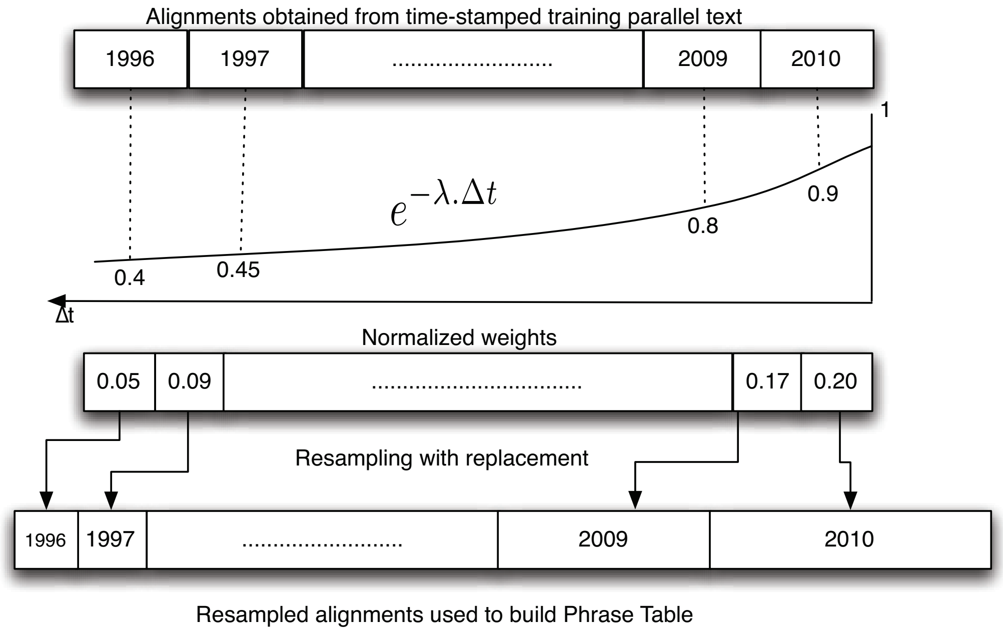

Figure 4.1: Overview of the weighting scheme. The alignments are weighted by an exponential decay function, parametrized by λ. Resampling with replacement is used to create a new corpus (parts with higher weight will appear more often). The phrase table is built from this corpus using the standard procedure.

where λ is the decay factor and Δt is the discretized time distance (0 for most recent part, 1 for the next one, etc.). Therefore, our weighting scheme has only one parameter to be optimized.

Following our work described in previous chapter we resample the alignments in order to obtain a weighting of the bitexts according to their recency. The weight of each part of the bitexts is normalized (sum to one). The normalized weights represent the percentage of the final aligned corpus that is originated from each part of the source corpus: word alignments corresponding to bitexts that are close to the test period will appear more often than the older ones in the final corpus.

In addition, we considered the quality of the alignments during resampling, as described in section 3.3.2 of chapter 3.

$$\log(\alpha \cdot \frac{(\sqrt[n_{trg}]{a_{src\_trg}} + \sqrt[n_{src}]{a_{trg\_src}})}{2}) \qquad (4.2)$$

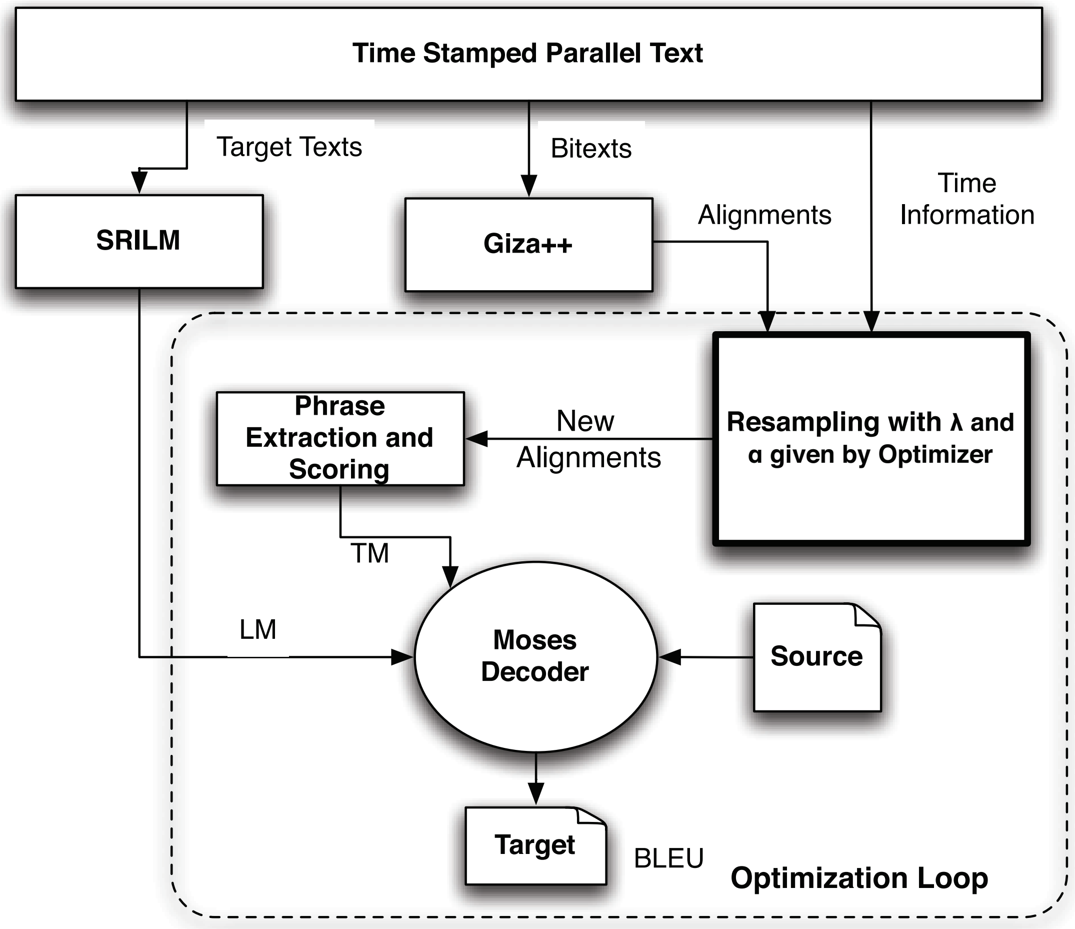

Figure 4.2: Architecture of SMT Weighting System.

where a is the alignment score, n the size of a sentence and α a smoothing coefficient to optimize. We used the normalized alignment scores as confidence measurement for each sentence pair.

4.3 Description of the algorithm

The architecture is presented in Figure 4.2. The starting point is a parallel corpus. We performed word alignment in both directions using GIZA++. The corpus is then separated into several parts on the basis of a given time span. We performed experiments with different span sizes, namely year, month, week and day. The

Algorithm 3 Weighting with Exponential Decay function using resampling

```
Determine word to word alignment with GIZA++ on concatenated bitexts.
Initialize $\lambda$ and $\alpha$ with equal weights.
while not Optimized do
    Compute time-spans weights by eq. 4.1
    Normalize weights
    for $i = 0$ to #time-span do
        $proportion$ ← required_size[1] ∗ weights[$i$]
        j = 0
        while $j < proportion$ do
            $Al$ ← Random alignment
            $Al_{score}$ ← normalized score of $Al$
            Flatten $Al_{score}$ with $\alpha$
            $Threshold \leftarrow$ rand[0, 1]
            if $Al_{score} > Threshold$ then
                keep it
                $j = j + 1$
            end if
        end while
    end for
    Create new resampled alignment file.
    Extract phrases and build the phrase table.
    Decode
    Calculate the BLEU score on Dev
    Update $\lambda$ and $\alpha$
end while
```

decay function is scaled so that the range does not change when using different span sizes. A weighting coefficient obtained with the exponential decay function is associated to each part.

Then, for each part, resampling with replacement is performed in order to select the required number of alignments and form the final corpus. We follow the same algorithm for resampling as described in section 3.3 of chapter 3.

Note that some alignments may appear several times, but this is exactly what is expected as it will increase the probability of certain phrase pairs which are supposed to be more related to the test data (in terms of recency) and of better

[1] required_size depends upon the number of times we resample - see section 5.

quality. The smoothing and decay factors, α and λ respectively, are optimized with a same numerical optimizer CONDOR Berghen and Bersini [2005] as mentioned in previous chapter. The procedure and steps involved in our weighting scheme are shown in algorithm 3.

4.4 Experimental evaluation

Our first experiments are based on the French-English portion of the freely available time-stamped Europarl data Koehn [2005] from April 1996 to December 2010. We have built several phrase-based systems using the Moses toolkit Koehn et al. [2007b], though our approach is equally applicable to any other approach based on alignments and could be used for any language pairs.

In the first experiments, the whole Europarl corpus was split into train, development and test as shown in Figure 4.3. The most recent 5K sentences are split into two sets of equal size, one for development and the other for testing. The remaining data was used as training bitexts to build the different systems.

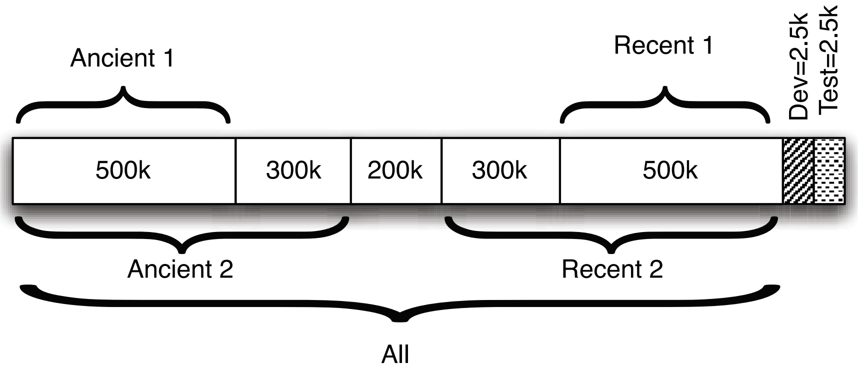

Figure 4.3: Data used to build the different systems (# sentences)

Since we want to focus on the impact of the weighting scheme of the bitexts, we used the same language model for all systems. It has been trained with the SRILM toolkit Stolcke [2002] on the target side of all the training data. In addition, the weights of the feature functions were tuned once for the system that uses all the training data and then kept constant for all the subsequent

Europarl	Ancient data		Recent data		
	Ancient 1	Ancient 2	Recent 1	Recent 2	All
# of sentences/words	500K/15M	800K/25M	500K/15M	800K/24M	1800K/55M
BLEU (on dev)	29.84	30.08	30.80	31.09	**31.34**
BLEU (on test)	29.30	29.43	30.32	30.44	**30.48**

Table 4.1: BLEU scores obtained with systems trained on data coming from different time spans.

experiments, *i.e.* no tuning of the feature functions weights is done during the optimization of the weighting coefficients λ and α.

Table 4.1 presents the results of the systems trained on various parts of the available bitexts without using the proposed weighting scheme. The best performance is obtained when using all the data (55M words, BLEU=30.48), but almost the same BLEU score is obtained by using only the most recent part of the data (24M words, part *Recent 2*). However, if we use the same amount of data that is further away from the time period of the test data (25M words, part *Ancient 2*), we observe a significant loss in performance. These results are in agreement with the observations already described in Levenberg et al. [2010]. Using less data, but still close to the evaluation period (15M words, part *Recent 1*) results in a small loss in the BLEU score. The goal of the proposed weighting scheme is to be able to take advantage of all the data while giving more weight to recent data than to older one. By these means we are not obliged to disregard older parts of the data that may contain additional useful translations. If the weighting scheme does work correctly, we cannot perform worse than using all the data. Of course, we expect to achieve better results by finding the optimal weighting between recent and ancient data.

The amount of data per year in the Europarl data can vary substantially in function of time period since it depends on the frequency and length of the sessions of the European Parliament. As an example Figure 4.4 shows the histogram of the data per year.

One can ask which time granularity should be used to achieve best weights. Only one parametrized weight is given to each time span, consequently the span

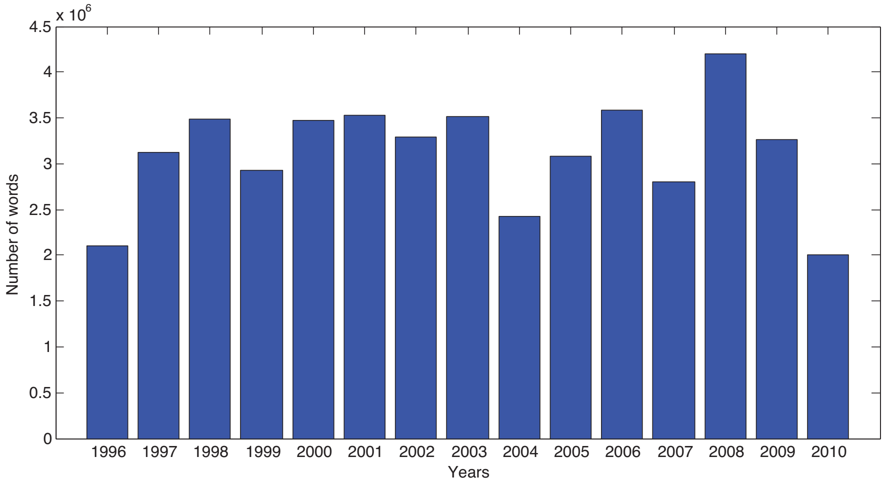

Figure 4.4: Amount of data available in the Europarl corpus for each year

Europarl	Weighting + alignment selection				Best+retune
Time span	Days	Weeks	Months	Years	Years
Optimized λ	0.0099	0.0109	0.0110	0.0130	0.0130
BLEU (on dev)	31.73	31.82	31.75	31.80	**31.92**
BLEU (on test)	30.94	30.97	30.92	**30.98**	**31.09**

Table 4.2: Results in BLEU score after weighting.

size will have an impact on the alignment selection process. Using smaller spans results in a more fine grained weighting scheme. We have tested different settings with different time spans to see whether the impact of weighting changes with the size of each span. The results are shown in Table 4.2.

It is observed that all four systems obtained very similar results, which indicates that the size of the spans is not very important. One surprising observation is that the optimized decay factor for all time span sizes are really close to each other. The reason to this could be the scaling of the exponential decaying function based on the time span size. In fact scaled values ensure that the oldest data point get roughly the same value independent of using years, months or days as time span. Looking at the optimized values of λ in Table 4.2, we can observe that the relative difference between recent and ancient data is rather small, *i.e.*

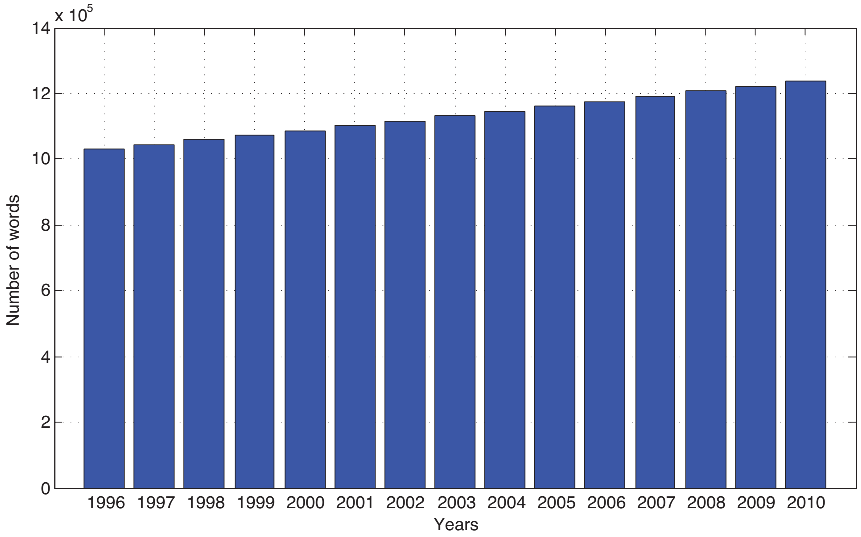

Figure 4.5: Distribution of data after weighting

the ancient data is still somehow important and cannot be neglected.

By using years as time span, we obtain an improvement of +0.50 BLEU score on the test set compared to using all data without weighting (30.48 → 30.98). It is clear that recency has a positive impact on system performance, however, weighting properly the different parts gives better performance than using the most recent or all available data.

Finally, the best system is retuned (feature functions weights) and an overall improvement of +0.61 in the BLEU score is observed on test set.

4.5 Discussion

The optimal decay factor of approximately 0.01 actually leads to an almost linear decrease over time. The difference in the quantity of data taken from most recent and least recent data is only 1.4% (which still represent 200k sentences). Therefore, one could think that the weighting does not favor recent data that much. This is not the case as we can see in Figure 4.5 where the distribution of data used to build the adapted model is presented. When comparing it to Figure 4.4, the overall proportion of data coming from recent years is clearly

Europarl	Resampling only	Weighting only	alignment selection only
BLEU (on dev)	31.36	31.69	31.45
BLEU (on test)	30.51	30.84	30.64

Table 4.3: Results in BLEU score with different settings.

bigger when using our resampling approach. This leads to different word choices while decoding.

Note that resampling is performed several times to estimate and select the samples which better represent the target data set. The more often we resample, the closer we get to the true probability distribution. The *required-size* in algorithm 3 depends upon the number of times we resample. We resampled ten times in our experiments. It is also worth to note that, we keep the original training data along with resampled one. It ensures that no information is lost and the set of extracted phrase pairs remain the same - only the corresponding probability distributions in the phrase table are changed.

In order to get more insight in our method, we separately performed the different techniques:

- resampling the training data without weighting;
- resampling the training data using weighting only (with respect to recency);
- resampling the training data using alignment selection.

These results are summarized in Table 4.3.

Note that the first case does not correspond to duplicating the training data a certain amount of time (which would of course produce exactly the same phrase-table). Since we perform resampling with replacement, this procedure introduces some randomness which could be beneficial. According to our results, this is not the case: we obtained exactly the same BLEU scores on the dev and test data than with the standard training procedure. Weighting with respect to recency or alignment quality both slightly improve the BLEU, but not as much as both techniques together. The performance increase seems actually to be complementary.

Example 1	
A:	Mr Ribeiro e Castro, we **shall** see all this in the Conference of Presidents.
B:	Mr Ribeiro e Castro, we **will** see all this at the Conference of Presidents.
R:	Mr Ribeiro e Castro, we **will** look at all these questions in the Conference of Presidents' meeting.
Example 2	
A:	We **shall** most probably consider again lodge a complaint with the **Court of Justice of the European Communities**.
B:	We **will** most probably consider again to lodge a complaint to the **European Court of Justice**.
R:	Most probably we **will** again discuss renewed recourse to the **European Court of Justice**.
Example 3	
A:	no Member State has *not* led to **field trials** as regards the BST .
B:	no Member State has led to **tests on the ground** as regards BST .
R:	No Member State has yet carried out **field tests** with BST .

Table 4.4: Example translations produced by systems *All* (A) and *Best+retune* (B) versus reference (R)

Some comparative examples between the translations produced by systems *All* and *Best+retune* versus the reference translations are given in Table 4.4. It was noticed that a lot of occurrences of *"will"* in the reference are actually translated into *"shall"* with system *All* whereas the correct word choice is made by the system *Best+retune* as shown in Example 1. This could be explained by the fact that recently the word *"will"* is more frequently seen in the training corpus and adapting the model by weighting the most recent data produced correct translation. Actually, it was found that the word *"will"* is 10% more frequent in recent data (*Recent 1*) than in ancient data (*Ancient 1*) while the word *"shall"* is 2% less frequent.

Another interesting example is Example 2, in which the correct name for the *European Court of Justice* is proposed by the adapted system unlike the system *All* which proposed *Court of Justice of the European Communities*. Actually, it appears that the *Court of Justice of the European Communities* is the former name of the *European Court of Justice* prior to December 2009. The use of recent data allows to correctly translate the named entities which can change over time. The correct translation proposed by System *Best+retune* could be observed in Example 3 because of the alignment selection procedure.

In our experiments, we assume that the test data is in present time (the usual case in a news translation system), and consequently we decrease the weight of the bitexts towards the past. This principle could be of course adapted to other scenarios.

An alternative approach could be to directly use the time decay function as the count for each extracted phrase. However, resampling the alignments and changing the counts of extracted phrases is not exactly the same. Same phrase pairs could be extracted from different parallel sentences coming from different time spans. Furthermore, weighting the alignments with their scores has shown improvements in the BLEU score as presented in Table 4.3, but considering the alignment score at the phrase level is not straight forward.

4.6 Experiments on the WMT task

To further verify whether our results are robust beyond the narrow experimental conditions, we considered a task where the development and test data do not come from the same source than the bitexts. We took the official test sets of the 2011 WMT translation tasks as dev and test sets Schwenk et al. [2011], *i.e.* news-test09 and news-test10 respectively. We built English-French systems by using the Europarl and News-Commentary (NC) corpora, both contain news data over a long time period.

For this set-up, there are three coefficients to optimize: the decay factor for Europarl λ_1, the decay factor for the news-commentary texts λ_2 and a coefficient for the alignments α. The Europarl corpus was divided into time span according to years and *NC* corpus was assumed to be sorted over time since time-stamp information was not available for the *NC* corpus. Remaining settings are kept the same as mentioned in previous experiments to build the system *Best+retune*. The results are shown in Table 4.5. Finally, we considered the relative importance of the Europarl and *NC* corpora. For this, a weight is attached to each corpus which represents the percentage of the final aligned corpus that comes from each

WMT Task	Baseline	WWR + AS	WWR + AS + RI
BLEU (on dev)	26.08	26.51	**26.60**
BLEU (on test)	28.16	28.59	**28.69**

Table 4.5: Results in BLEU score after weighting on English to French WMT Task. WWR=Weighting With Recency, AS=Alignment Selection, RI=Relative Importance

source corpus. These weights are also optimized on the development data using the same technique as proposed in our previous chapter. Using all these methods, we have achieved an overall improvement of approximately +0.5 BLEU on the development and test data, as shown in Table 4.5.

4.7 Conclusion

In this chapter, a parametric weighting technique along with resampling is proposed to weight the training data of the translation model of an SMT system. By using a parametric weighting function we circumvented the difficult problem to numerically optimize a large number of parameters. Using this formalism, we were able to weight the parallel training data according to the recency with respect to the period of the test data. By these means, the system can still take advantage of all data, in contrast to methods which only use a part of the available bitexts. We evaluated our approach on the Europarl corpus, translating from French into English and further tested it on official English to French WMT Task. A reasonable improvement in BLEU score on the test data was observed in comparison to using all the data or only the most recent one. We argue that weighting the training data with respect to its temporal closeness should be quite important for translating news material since word choice in this domain is rapidly changing.

An interesting continuation of this work is to consider other criteria for weighting the corpora than the temporal distance. It is clear that recency is a relevant information and this could be associated with other features, *e.g.* thematic or linguistic distance. Also, this work can be included into a stream-based framework where new data is incorporated in an existing system by exponential growth function and making use of online retraining procedure as discussed in Levenberg et al. [2010].

Chapter 5

A General Framework

5.1 Background

The approaches discussed in chapter 3 and 4 to adapt the SMT models by resampling the data has shown positive results. In spite the fact, those improvements are quite reasonable; there is an issue about the practical execution of the approach. Generally data is resampled many times for better data distribution, which results into better estimations. Secondly tuning of the sample coefficients is needed to adjust the weights for each sample. More sample groups in the data results into more coefficients to optimize. An SMT model trained on huge amount of data takes great deal of time. Resampling many times introduces the overhead of bigger training data and rebuilding the models in many iterations to optimize the sample weights makes the procedure incredibly long.

5.2 Overview of idea

The framework proposed in this chapter is motivated along two axes. Firstly to coup with the problem of time effectiveness and secondly to suppress the sample bias problem on a smaller granularity like phrases at later stages of translation model training. The technique is based on shifting the balance of probability estimation from biased sample to the domain of interest at scoring step. The size of the data remains the same as original.

The proposed framework is an extension and generalization of several ideas proposed in previous chapters and to investigate weighted counts with feature scores. Our proposed framework gives the flexibility to inject the feature scores in a unified formulation calculated at various levels. It is based on the following principles:

- the use of a set of "quality measures" at different levels: weights for each corpus (or data source) and for each individual sentence in the bitexts.
- no additional feature functions to express the quality or appropriateness of certain phrase pairs, but we modify only the existing phrase probabilities. By these means, we don't have to deal with the additional complexity of decoding and optimizing the weights of many feature functions.
- resampling the bitexts or alignments is computationally expensive for large corpora since the resampled data is ideally much bigger than the original one. Instead, we integrate the various weighting schemes directly in the calculation of the translation probabilities.
- our approach has only a small number of parameter to optimize.

The rest of the chapter is organized as follows. In the next section we present in detail the architecture of our approach. Experimental results for IWSLT'11 and WMT'11 task are summarized and discussed in section 5.4. The chapter concludes with a discussion and perspectives of this work.

5.3 Architecture of our approach

In our approach we only modify the way how the phrase translations probabilities $P(\tilde{e}|\tilde{f})$ and $P(\tilde{f}|\tilde{e})$ are calculated. The goal is to increase the probability of phrase pairs which are more important or reliable for the considered task, and consequently, to down weight those which should be used less often. It is important to point out that our phrase table has exactly the same number of entries than the original one and that we do not add more scores or feature functions. Currently, we do not modify the lexical scores of each phrase pair, but we will investigate this in the future. In summary, we only modify step 7 in the procedure described in section 3.2. For this we modified the tool `memscore` Hardmeier [2010].

In practice, we also need to adapt step 5 since we need to keep track for each phrase pair , the corpus it was extracted from and the scores of the corresponding sentence.

5.3.1 Standard phrase probabilities

The standard procedure to calculate the phrase probabilities is simple relative frequency:

$$P(\tilde{e}_{ij}|\tilde{f}_i) = \frac{Count(\tilde{f}_i, \tilde{e}_{ij})}{\sum_k Count(\tilde{f}_i, \tilde{e}_{ik})} \tag{5.1}$$

The `memscore` tool also implements various smoothing methods such as Witten-Bell, Kneser-Ney discounting etc. but to the best of our knowledge, their eventual benefit was not extensively studied and these smoothing techniques are not widely used. In any case, the calculation of the phrase probabilities does not consider from which corpus the phrase was extracted, or more generally, any kind of weight that was attached to the originating sentence.

This can obviously lead to wrong probability distributions. As a simple example we can consider a phrase pair $\tilde{f}_i, \tilde{e}_{ij}$ which appears a couple of times in the in-domain corpus, and which provides the correct translation for the task, and another phrase pair $\tilde{f}_i, \tilde{e}_{ik}$ which appears many times in a (larger) out of-domain corpus. This wrong translation will wrongly get a higher probability when relative frequency estimates are used (or any of the standard smoothing techniques).

A similar argumentation holds at the sentence or even phrase level. For instance, even a generally in-domain corpus can contain few sentences which are out-of topic or badly aligned.

5.3.2 Weighted phrase probabilities

We have modified the `memscore` tool in order to take into account a weight attached to each corpus and let us assume that we have the following information on our parallel training data:

- the parallel data can be organized into C different parts. In most of the cases, we will use the source of the data to partition it, *e.g.* Europarl, United Nations, web-crawled, but one could also use some kind of clustering algorithm. We associate the weight w_c, to each corpus c=1 $\ldots C$. We will discuss later how to obtain those weights.

- a set of S "goodness scores" $q_s(f_i, e_i), s = 1 \ldots S$ for each parallel sentence pair $(f_i, e_i), i = 1 \ldots L$ where L is the number of parallel sentences. Again, we will delay for now how to produce those sentence scores. We keep track of these sentence scores when extracting phrases. All the phrases extracted from the same sentence obtain the same phrase-level goodness scores $h_s(f_j, e_j), j = 1 \ldots P$ where $P \gg S$ is the number of extracted phrases.

Using these notations we will calculate the phrase probability as follows. Let us first consider only the weights of the individual corpora. This is achieved by extending equation 5.1 as follows:

$$P(\tilde{e}_{ij}|\tilde{f}_i) = \frac{\sum_{c=1}^{C} w_c Count_c(\tilde{f}_i, \tilde{e}_{ij})}{\sum_{c=1}^{C} w_c \sum_k Count_c(\tilde{f}_i, \tilde{e}_{ik})} \tag{5.2}$$

The equation 5.2 is identical as given in Matsoukas et al. [2009], where w_c represents the features-mapped to a weight calculated for each sentence by neural network. However in our case it represents the direct weight for each corpus. If all corpus weights are identical, equation 5.2 simplifies to the original formulation in equation 5.1. Considering in addition the goodness scores at the sentence level, we will get:

$$P(\tilde{e}_{ij}|\tilde{f}_i) = \frac{\sum_{c=1}^{C} w_c Count_c(\tilde{f}_i, \tilde{e}_{ij}) \cdot \prod_{s=1}^{S} \gamma_s h_{c,s}(f_i, e_{ij})}{\sum_{c=1}^{C} w_c \sum_k Count_c(\tilde{f}_i, \tilde{e}_{ik}) \cdot \prod_{s=1}^{S} \gamma_s h_{c,s}(f_i, e_{ik})} \tag{5.3}$$

where γ_s is an additional parameter to weight the different sentence goodness scores among each other. We implemented phrase probability calculation according to equation 5.3 in the `memscore` tool of Moses.

5.3.3 Calculation of the corpus weights and sentence features

Our theoretical framework and implementation is generic and does not depend on the exact calculation of the corpus weights or the sentences goodness scores. Any value that expresses the appropriateness of the corpus and sentence with respect to the task can be used. In the following we outline some possibilities which were used in our experiments.

Weighting parallel corpora was already investigated in previous chapters, where we used a resampling technique to weight parallel corpora. We have proposed two methods to obtain the corpus weights: via LM interpolation and numerical optimization to maximize the BLEU score on some development data. The second approach showed slightly better performance, but it is computationally quite expensive (a new phrase table must be build for each optimization loop). Therefore, we decided to use corpus weights obtained by LM interpolation in our experiments. The idea is to build a LM on the source (or target) side of the bitexts, independently for each corpus. The resulting corpus coefficients can be directly used to weight the parallel corpora.

Perplexity can also be used to weight each individual sentence. This was used to select a relevant subset of LM data [Axelrod et al., 2011] or bitexts [Moore and Lewis, 2010]. In our case, we build a LM on the source side of the in-domain corpus and use this model to calculate the perplexity of each sentence in all the other corpora. Since lower perplexity represents "better" sentences, we set $q(s_i, t_i)$ to the inverse of the perplexity. It is important to note that our approach is a generalization of data selection approaches: instead of doing a hard decision which data to keep to discard, we keep all the sentences and attach a weight to each one (this weight could be zero in an extreme case).

It was also observed that parallel sentences which are closer to the test set period are more important than older ones [Hardt and Elming, 2010; Levenberg et al., 2010; Shah et al., 2011], in particular when translating texts in the news domain. Following our work described in section 4.1 of chapter 4, we use an exponential decay function or this goodness function:

$$q(f_i, e_i) = e^{-\alpha \cdot \Delta t} i \tag{5.4}$$

where α is the decay factor and Δt is the discretized time distance (0 for most recent part, 1 for the next one, etc.).

Finally, it was argued in previous chapters that the alignment score produced by Giza++ could be used as a measure whether the phrases extracted from the corresponding sentence pair should be up- or down-weighted. In order to ease comparison, we used the same equation as described in previous chapters:

$$q(f_i, e_i) = \log(\beta \cdot \frac{(\sqrt[n_{trg}]{a_{src\_trg}} + \sqrt[n_{src}]{a_{trg\_src}})}{2}) \tag{5.5}$$

where a is the alignment score, n the size of a sentence and β a smoothing coefficient to optimize.

5.3.4 Overall architecture

The overall architecture of our approach is given in figure 5.1. Suppose we have several parallel corpora coming from various sources. First of all, sentence level features are calculated and synchronized with the parallel sentences. These sentences are concatenated to perform word-to-word alignment in both directions using GIZA++. Alignment scores corresponding to each sentence pair are added to the feature file. Then, phrases are extracted and the goodness score $q(s_i, t_i)$ is synchronized with the phrases. Finally, the phrase-translation probabilities are calculated according to equation 5.3 in forward and backward direction.

The parameters of our approach α , β and γ along with w_c are numerically optimized. In this optimization loop we keep the weights of the feature functions constant, *i.e.* λ_i in equation 3.1 (we use the ones of the standard system without weighted phrase translation probabilities). Eventually, these weights are optimized using the standard MERT procedure once we have fixed the parameters of our approach.

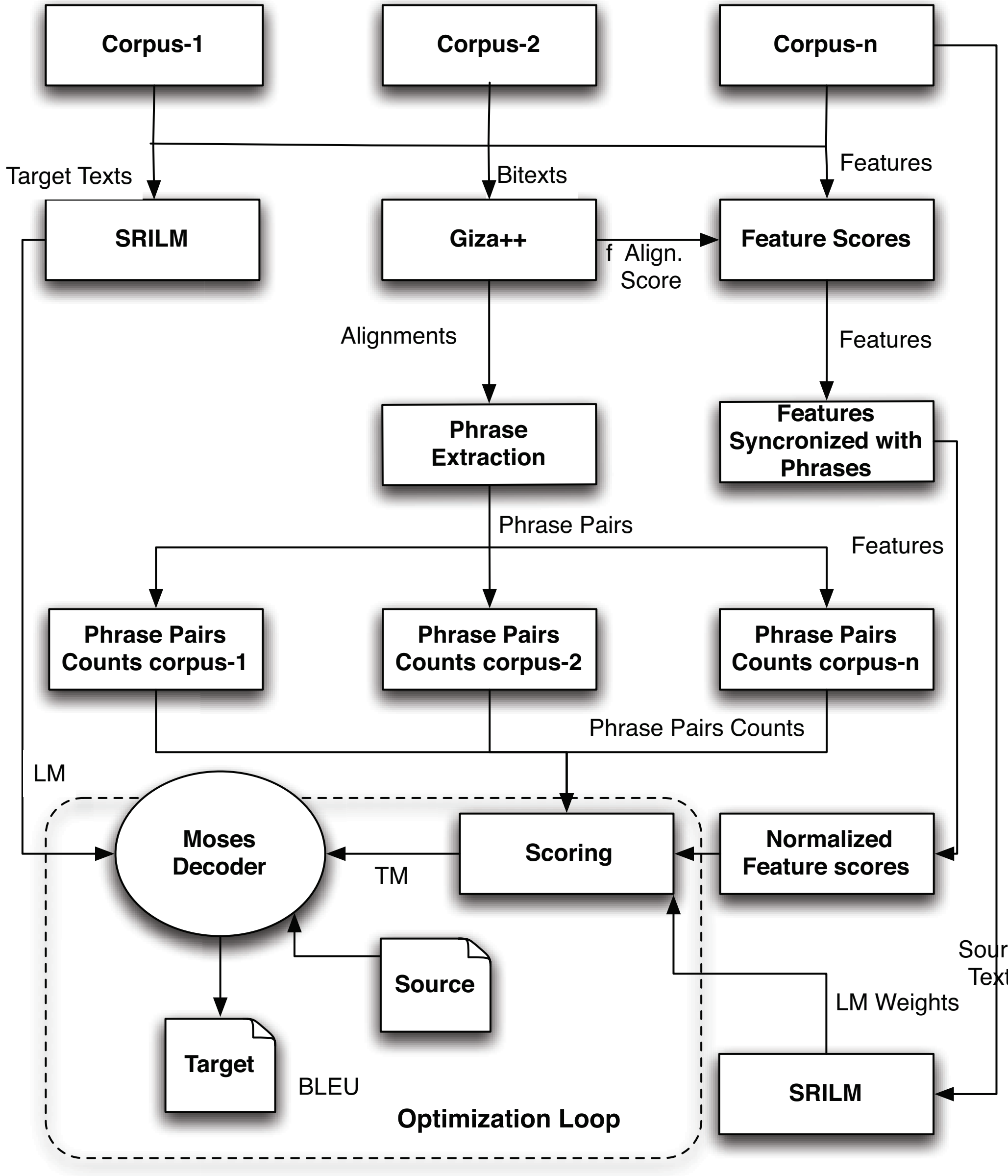

Figure 5.1: Overall architecture of our approach.

5.4 Experimental evaluation

We have built several phrase-based systems using the Moses toolkit Koehn et al. [2007b]. The scoring framework is implemented by extending the memory based scoring tool called `memscore` Hardmeier [2010] available in the Moses toolkit.

Corpus	En tokens	Fr tokens
TED	2.0	2.2
News-Commentary	2.8	3.3
Europarl v6	50.6	56.2
ccb2	232.5	272.6
TOTAL	287.9	334.3

Table 5.1: Size of parallel corpora (in millions) to build baseline systems for WMT and IWSLT Tasks.

WMT Task	Corpus weights	Alignment scores	Temporal distance	perplexity	BLEU (on test)
Baseline					28.16
System 1	yes				28.41
System 2		yes			28.21
System 3			yes		28.35
System 4				yes	28.56
System 5	yes	yes			28.55
System 6	yes		yes		28.60
System 7	yes			yes	28.61
System 8			yes	yes	28.79
System 9	yes	yes	yes		28.65
System 10	yes	yes		yes	28.67
System 11	yes		yes	yes	**28.89**
System 12	yes		yes	yes	**29.11 (optimized)**

Table 5.2: BLEU scores obtained with systems trained with different features on WMT Task.

The experiments are performed on two well-known evaluation tasks, *i.e.* the 2011 WMT and IWSLT English/French evaluations. The corpora and their sizes used to build the systems for both these tasks are given in table 5.1.

5.4.1 Experiments on the WMT task

For the WMT task we used the official development sets of the 2011 WMT translation tasks, *i.e* news-test09 as development corpus and news-test10 as test corpus. We built English-French systems by using the time-stamped *Europarl* and news-commentary (*nc*) corpora. The LM is created by interpolating several language models trained separately on the target side of the bitexts and all available target language monolingual data (about 1.5G words). These individual language models are interpolated and merged into one huge model. The coefficient of the individual models are optimized using the usual EM procedure to minimize perplexity on the development data. Initial corpus weights for the bitexts were obtained by building another interpolated LM on the target side of the bitexts only.

We explored the following features to weight the relevance of the bitexts and the individual sentences: corpus weights, alignment scores, recency of the data with respect to the test set period and the sentence perplexity in the target language with respect to an in-domain language model. The news-commentary (*nc*) corpus was used for that purpose. The time information provided with *Europarl* data is used to estimate recency feature. This information was not available for *nc*, so we considered the sentences in chronologically ordered with respect to temporal distance. The alignment scores provided by GIZA++ were normalized using equation 5.5.

The results of the baseline system and various combinations of the different feature functions are summarized in table 5.2. In order to get an idea which feature functions give best results, we have first performed experiments using default values for the parameters of the feature functions. For this purpose, we have used the values reported to be optimal in previous chapter.

The baseline system achieves a BLEU score of 28.16 on the test set. Each feature functions alone brought small improvements in the BLEU score (systems 1–4 in Table 5.2), the best being sentence perplexity (+0.4 BLEU). An interesting property of our approach is that the individual gains seem to add up when we use several feature functions, for instance combining recency and sentence perplexity gives and improvement by 0.63 BLEU (system 8) while the individual improvements are only +0.19 and 0.40 respectively. Combining corpus weights and sentence perplexity is less useful, as expected, since sentence perplexity implicitly weights the corpora. This is in fact an improved corpus weighting with a finer granularity. Our best system was obtained when combining corpus weights, recency and sentence perplexity weighting (system 11). For this system only,

IWSLT Task	Corpus weights	Alignment scores	perplexity	BLEU (on test)
Baseline				26.34
System 1	yes			26.61
System 2		yes		26.41
System 3			yes	26.77
System 4	yes	yes		26.51
System 5	yes		yes	**26.86**
System 6	yes		yes	**26.99 (optimized)**
System 7		yes	yes	26.81
System 8	yes	yes	yes	**26.91**
System 9	yes	yes	yes	**27.07 (optimized)**

Table 5.3: BLEU scores obtained with systems trained with different features on IWSLT Task.

we numerically optimized the weights w_c, α and γ on the development set (see figure 5.1). The default and new weights are:

$$\begin{aligned}
w_{eparl} &= 0.47714 \rightarrow 0.32823 \\
w_{nc} &= 0.52285 \rightarrow 0.67121 \\
\alpha &= 0.01300 \rightarrow 0.02102 \\
\beta &= 0.14530 \rightarrow 0.12901 \\
\gamma_{as} &= 0.1 \rightarrow 0.01289 \\
\gamma_{td} &= 0.1 \rightarrow 0.19201 \\
\gamma_{ppl} &= 0.1 \rightarrow 0.15451
\end{aligned}$$

where γ_x is the coefficient among alignment score (as), temporal distance(td) and perplexity(ppl). By these means, we get an overall improvement of roughly +1 BLEU score (28.16 → 29.11) on test set. It is important to stress that this system is trained on exactly the same data than the baseline system and that the phrase table contains the same phrase-pairs. Our approach only improves the forward and backward probability estimates $P(\tilde{t}|\tilde{s})$ and $P(\tilde{s}|\tilde{t})$.

IWSLT Task	c_{ted}	c_{nc}	$c_{ccb2.px70}$	β	γ_{ppl}	γ_{as}
Default values	0.74032	0.17378	0.08591	0.1	0.1	0.1
Optimized	0.69192	0.16982	0.13831	0.19251	0.18151	0.03118

Table 5.4: Feature weights on IWSLT Task (ppl=perplexity, as=alignment score).

5.4.2 Experiments on the IWLST task

We performed the same type of experiments for the IWSLT task. The parallel training data was the in-domain *TED* corpus, the news-commentary corpus (*nc*) and a subset of the French–English 10^9 Gigaword (internally called *ccb2*). The results for this task are summarize in Table 5.3. The official Dev and test sets of the 2011 IWSLT talk task are used. Initial experiments have shown that large parts of the *ccb2* corpus are not relevant for this task (looking at the sentence perplexities). Therefore, we decided to only use a subset of this corpus, namely all the sentences with a perplexity lower than 70. This process preserve only 3% of the *ccb2* data. The baseline system trained on this data achieves a BLEU score of 26.34 on the test data. Using all the data in *ccb2* worsens significantly the results: the BLEU scores is 25.73. In principle, it is not necessary to select subsets of the parallel training data with our approach to weight sentences by perplexity, but this speeds up processing since we do not need to consider many sentences pairs with a very low weight. We perform a kind of pruning: all those sentences get a zero weight and are discarded right away. We used the LM build on the in-domain *TED* to calculate the sentence perplexities and the LM interpolation weights are used as corpus weights. The recency feature was not used for this task since the test set of the TED corpus has no time information.

We observed the same behavior than for the WMT task: each individual feature function improves the BLEU score on the test set (systems 1–3), weighting by sentence perplexity being the best one (+0.43 BLEU). The best system is obtained when combining all three feature functions, leading a BLEU score of 26.91 (system 8). Again, the numerical optimization of the weights of the feature functions achieves an additional small improvement, giving an overall improvement of 0.73 BLEU. The weight of the feature functions are shown in table 5.4.

5.5 Comparative Analysis

It is interesting to compare the impact of the different features considered. An interesting fact is that the trend is similar for both tasks.

By comparing the results obtained with the various systems, we can observe that the **corpus weights**, used alone or in combination with other features, are always beneficial (by pairwise comparison of *e.g.* systems 2 and 5, systems 3 and 6 or systems 4 and 7 from WMT task). The average gain provided by such weighting is around 0.2. Those weights correspond to the LM interpolation coefficient optimized to minimize the perplexity on the development set. They are useful to weight a whole corpus and to ensure that the in-domain corpus will globally receive higher weight than the other corpora.

Sentence level perplexity is also always useful (compare *e.g.* systems 1 and 7 or systems 6 and 11 from WMT task). While one could think that this feature is redundant with corpus weight, it does bring additional information about the relevance of the sentence. This can be explained by the fact that a globally out-of-domain corpus can contain a fraction of useful sentences while, on the contrary, an in-domain corpus may contain some less useful ones. This is part of the heterogeneous aspect of any corpus. The average gain of using sentence perplexity is almost 0.3 for the WMT task and 0.37 for IWSLT task.

Concerning the **alignment score**, the results obtained are more mitigated (see *e.g.* the comparison between systems 2 and 5 on WMT and systems 2 and 4 from IWSLT task). The average gain is very low, and it is the only feature which sometimes decrease the BLEU score. The **temporal distance** has the expected behavior. When comparing systems 1 and 6, 3 and 8 or 4 and 11 from WMT Task, we can observe that an improvement of more than 0.2 is obtained.

5.6 Conclusion

We have proposed a general framework to improve the phrase translation probabilities in a phrase-based SMT system. For this, we use a set of "goodness scores" at the corpus or sentence level. These feature functions are used to calculate forward and backward phrase translations probabilities which are better adapted to the task. Our framework and implementation is generic and does not depend on the exact calculation of the corpus weights or the sentences goodness scores. Any value that expresses the appropriateness of the corpus and sentence with respect to the task can be used. The adapted system has exactly the same time and space complexity than the baseline system since we do not modify the number of entries in the phrase-table or add additional features. Also, the training time is only slightly increased.

We evaluated this approach on two well-known tasks: the 2011 WMT and IWSLT English/French evaluations. We have investigated several feature functions: weights for corpora coming from different sources and weights at the sentence level based on the quality of the GIZA++ alignments, the recency with respect to the test set period and task appropriateness measured by the perplexity with respect to an in-domain language model. Using each one of these feature functions, improved the BLEU score with respect to a strong baseline. However, best results were obtained by using all the feature functions. This yielded an overall improvement of almost 1 point BLEU for the WMT task and more than 0.7 BLEU on the IWSLT task.

Chapter 6

Conclusions and future prospectives

In this thesis, we have proposed new techniques to automatically adapt the translation model of an SMT system. In contrast to many other works, we did not try to collect additional resources (bitexts, comparable corpora, etc), but developed methods that aim to use the available resources in the best possible manner. Given the fact that parallel texts is a sparse resource and heterogeneous with respect to many factors, it is important to weight them for better translation quality. Since many years, the focus of research in domain adaptation has been data selection, *i.e.* extracting a subset of the training data that is considered more relevant for the given task. This has been proposed for monolingual data to build language models as well as for bilingual data to train translation models. Data selection performs a hard binary decision to select data and it is difficult to determine the optimal amount of data to select. Discarding too much data may result in information loss. Weighting the data offers an attractive way to give more importance to relevant data for a given task by assigning appropriate weights to each data subset without discarding any data.

We have introduced multi-level weighting schemes in order to adapt the translation model. These schemes allow the integration of several "quality scores" at various levels during the creation of an SMT system. These scores include: corpus weights, sentences goodness scores like perplexity, temporal distance and alignment quality. These scores are considered in two ways: by resampling the data and by direct weighting of parameters. The first method was based on resampling the alignments, giving a weight to each corpus and using the GIZA++ alignment score as confidence measure of each aligned sentence pair. This technique does not change the phrase pairs that are extracted,[1] but only the corresponding probability distributions. By these means, we adapt the translation model in order to increase the weight of translations that are important to the task, and we down-weight the phrase pairs which result from unreliable alignments. We made no assumptions on how the phrases are extracted and it should be possible to apply the same technique to other SMT systems which rely on word-to-word alignments. We experimentally verified the new method on two very different tasks: IWSLT for which only a limited amount of resources are available, and NIST'08 OpenMT which is considered to have a large amount of training material for the translation model. We observed significant improvement on both tasks over state-of-the-art baseline systems. This weighting scheme is generic and can be applied to any language pair and target domain.

In an extended work, a parametric weighting technique along with resampling is proposed to weight the training data of the translation model of an SMT system. By using a parametric weighting function we circumvented the difficult problem to numerically optimize a large number of parameters. Using this formalism, we were able to weight the parallel training data according to the recency with respect to the period of the test data. By these means, the system can still take advantage of all data, in contrast to methods which only use a part of the available bitexts. We evaluated our approach on the Europarl corpus, translating from French into English and further tested it on the official English to French WMT Task. A good improvement in the BLEU score on the test data was observed in comparison to using all the data or only the most recent one. We argue that weighting the training data with respect to its temporal closeness should be quite important for translating news material since word choice in this domain is rapidly changing.

Finally, we directly use the weights of the corpora in the algorithm that extracts the phrase pairs and calculates their probabilities. This answered the interesting question whether resampling itself is needed or whether weighting the corpora and alignments is the key to the observed improvements in the BLEU

[1] when also including the original alignments

score. For this, we have used a set of *goodness scores* calculated at the corpus or sentence level. These goodness scores are used directly in the algorithm to calculate the phrase translation probabilities. The proposed framework is generic and does not depend on the exact calculation of the corpus weights or the sentences goodness scores. Any value that expresses the appropriateness of the corpus and sentences with respect to the task can be used. The adapted system has exactly the same time and space complexity than the baseline system since we do not modify the number of entries in the phrase-table or add additional features. We evaluated this approach on two well-known tasks: the 2011 WMT and the IWSLT English/French evaluations.

We reach to the following conclusion: Weighting the data is very important to improve the translation quality of SMT systems because of the heterogeneous nature of the data. Resampling the data showed promising results but at the cost of longer computational time. On the other hand, if the underlying phrase extraction and scoring algorithms are unknown, resampling is an ultimate choice. Direct weighting gives the flexible and efficient framework to integrate quality scores calculated at various levels directly into scoring framework.

6.1 Future prospectives

Our work on domain adaptation proposed in this thesis has many future prospectives. These may include:

- to explore other measures which predict the relevance of the training data to a given domain at various levels, *i.e.* corpus level, sentence level, alignment level or phrase level. These measure may include thematic or linguistic distance, topic closeness, cosine measure, etc.

- to integrate our proposed goodness scores directly into the log-linear framework as feature functions. By this mean we could try to analyze at which step of SMT system training, weighting perform best in term of time efficiency and translation quality.

- to use other objective functions than BLEU while optimizing the coefficients of the goodness scores

Appendix A

Publications

- Kashif Shah, Loïc Barrault, and Holger Schwenk. Translation model adaptation by resampling. In Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, WMT 10, pages 392399, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-71-8.

- Kashif Shah, Loïc Barrault, and Holger Schwenk. Parametric weighting of parallel data for statistical machine translation. In Proceedings of 5th International Joint Conference on Natural Language Processing, pages 13231331, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

- Holger Schwenk, Patrik Lambert, Loïc Barrault, Christophe Servan, Sadaf Abdul-Rauf, Haithem Ai, and Kashif Shah. Liums smt machine translation systems for wmt 2011. In Proceedings of the Sixth Workshop on Statistical Machine Translation, pages 464469, Edinburgh, Scotland, 2011. Association for Computational Linguistics

- Kashif Shah, Adaptation in statistical Machine Translation, journe des doctorants de l'école doctorale STIM (JDOC'10), Nantes, France.

- Kashif Shah, Loïc Barrault, and Holger Schwenk. A General Framework

to Weight Heterogeneous Parallel Data for Model Adaptation in Statistical Machine Translation (Submitted 2012)

Bibliography

Sadaf Abdul-Rauf and Holger Schwenk. On the use of comparable corpora to improve SMT performance. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 16–23, Athens, Greece, April 2009. 4, 36

D.J. Arnold, Lorna Balkan, Siety Meijer, R.Lee Humphreys, and Louisa Sadler. *Machine Translation: an Introductory Guide*. Blackwells-NCC, London, 1993. 11

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. 36, 72

F. Béchet, R. De Mori, and D. Janiszek. Data augmentation and language model adaptation using singular value decomposition. *Pattern Recogn. Lett.*, 25(1): 15–19, January 2004. ISSN 0167-8655. doi: 10.1016/j.patrec.2003.08.008. URL `http://dx.doi.org/10.1016/j.patrec.2003.08.008`. 32

Frank Vanden Berghen and Hugues Bersini. CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175, September 2005. 48, 59

Nicola Bertoldi and Marcello Federico. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the 4th EACL*

Workshop on Statistical Machine Translation, pages 182–189. Association for Computational Linguistics Association for Computational Linguistics, 2009. 35

Nicola Bertoldi, Mauro Cettolo, Roldano Cattoni, Boxing Chen, and Marcello Federico. Itc-irst at the 2006 tc-star slt evaluation campaign. In *TC-STAR Workshop on Speech-to-Speech Translation*, pages 19–24, 2006. 22

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 858–867, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 25

P. Brown, S. Della Pietra, Vincent J. Della Pietra, and R. Mercer. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2): 263–311, 1993. 16

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2): 79–85, June 1990. 16

Francisco Casacuberta and Enrique Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30:205–225, June 2004. 22

Francisco Casacuberta, Enrique Vidal, and Juan Miguel Vilar. Architectures for speech-to-speech translation using finite-state models. In *Proceedings of the ACL-02 workshop on Speech-to-speech translation: algorithms and systems - Volume 7*, S2S '02, pages 39–44, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 22

Mauro Cettolo, Marcello Federico, Nicola Bertoldi, Roldano Cattoni, and Boxing Chen. A look inside the itc-irst smt system. In *MT-Summit*, pages 451–457. Proc. of Machine Translation Summit X, 2005. 28

Eugene Charniak, Kevin Knight, and Kenji Yamada. Syntax-based language models for statistical machine translation. In *MT Summit IX. Intl. Assoc. for Machine Translation*, 2003. 27

Boxing Chen, Min Zhang, Aiti Aw, and Haizhou Li. Exploiting n-best hypotheses for SMT self- enhancement. In *Association for Computational Linguistics*, pages 157–160, 2008. 35

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-1996)*, pages 310–318, 1996. 25, 26

Stanley F. Chen, Kristie Seymore, and Ronald Rosenfeld. Topic adaptation for language modeling using unnormalized exponential models, 1998. 32

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 263–270, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. 22

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33:201–228, June 2007. 22

Jorge Civera and Alfons Juan. Domain adaptation in statistical machine translation with mixture modelling. In *Second Workshop on SMT*, pages 177–180, 2007. 34

Michael Denkowski and Alon Lavie. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*, 2011. 31

Thi-Ngoc-Diep Do. *Extraction de corpus parallèle pour la traduction automatique depuis et vers une langue peu dotée.* PhD thesis, Université de Grenoble, 2011. 4

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. 31

Kevin Duh, Katsuhito Sudoh, Tomoharu Iwata, and Hajime Tsukada. Alignment inference and bayesian adaptation for machine translation. In *Proceedings of Machine Translation Summit XIII, Xiamen, China*, 2011. 37

George Foster and Roland Kuhn. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135. Association for Computational Linguistics, 2007. 34

George Foster, Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 451–459, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1870658.1870702. 37

William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993. ISSN 0891-2017. 13

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 228–235, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1073012.1073042. URL http://dx.doi.org/10.3115/1073012.1073042. 27

Kevin Gimpel and Noah A. Smith. Generative models of monolingual and bilingual gappy patterns. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. 16

Christian Hardmeier. Fast and extensible phrase scoring for statistical machine translation. In *The Prague Bulletin of Mathematical Linguistics*, pages 87–96. Versita, Warsaw, 2010. 70, 74

Daniel Hardt and Jakob Elming. Incremental re-training for post-editing smt. In *The Ninth Conference of the Association for Machine Translation in the Americas 2010*, 2010. 54, 72

Masahiko Haruno and Takefumi Yamazaki. High-performance bilingual text alignment using statistical and dictionary information. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 131–138, Morristown, NJ, USA, 1996. Association for Computational Linguistics. 13

S. Hewavitharana, B. Zhao, A. S. Hildebrand, M. Eck, C. Hori, S. Vogel, and A. Waibel. The CMU Statistical Machine Translation System for IWSLT 2005. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2005. 22

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. Adaptation of the translation model for statistical machine translation based on information retrieval. In *EAMT*, pages 133–142, 2005. 35, 36

Fei Huang and Bing Xiang. Feature-rich discriminative phrase rescoring for smt. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 492–500, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1873781.1873837. 37

John Hutchins. Example-based machine translation: a review and commentary. *Machine Translation*, 19:197–211, December 2005. 13

W. John Hutchins and Harold L. Somers. *An Introduction to Machine Translation*. Academic Press, 1992. 10, 12

R. Iyer and M. Ostendorf. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *IEEE Transactions on Speech and Audio Processing*, pages 236–239, 1996. 32

Martin Kay and Martin Röscheisen. Text-translation alignment. *Computational Linguistics*, 19:121–142, March 1993. 13

Philipp Koehn. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. *Machine Translation: From Real Users to Research*, 3265:115–124, 2004. doi: 10.1007/978-3-540-30194-3_13. 27

Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT. URL http://mt-archive.info/MTS-2005-Koehn.pdf. 59

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ix, 21, 24

Philipp Koehn and Josh Schroeder. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227. Association for Computational Linguistics, 2007. 34

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. 19, 23, 41

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 *Meeting of the Association for Computational Linguistics*, pages 177–180, 2007a. 22, 27

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007b. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1557769.1557821. 49, 59, 74

Patrik Lambert, Holger Schwenk, Christophe Servan, and Sadaf Abdul-Rauf. Investigations on translation model adaptation using monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 284–293, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2132. 35

Alon Lavie and Abhaya Agarwal. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1626355.1626389. 31

Abby Levenberg, Chris Callison-Burch, and Miles Osborne. Stream-based translation models for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 394–402, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL http://portal.acm.org/citation.cfm?id=1857999.1858061. 54, 60, 67, 72

Adam David Lopez. *Machine Translation by pattern matching*. PhD thesis, Institute for Advanced Computer Studies, University of Maryland, 2008. 25

Yajuan Lu, Jin Huang, and Qun Liu. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages

343–350, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D/D07/D07-1036. 36

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. Spmt: Statistical machine translation with syntactified target language phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 44–52, Sydney, Australia, July 2006. Association for Computational Linguistics. 27

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 708–717, 2009. 37, 71

Evgeny Matusov, Zens R, Vilar D, Mauser A, Popovic M, Hasan S, and H. Ney. The RWTH machine translation system. In *TC-STAR Workshop on Speech-to-Speech Translation*, pages 31–36, 2006. 22

Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 220–224, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1858842.1858883. 72

Dragos Stefan Munteanu and Daniel Marcu. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4): 477–504, 2005. 4, 35

JA Nelder and R Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965. ISSN 1460-2067. doi: 10.1093/comjnl/7.4.308. 28

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 for Natural Language Processing*, 2004. 37

Franz Josef Och. *Statistical Machine Translation: From Single-Word Models to Alignment Templates.* PhD thesis, Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen, 2002. ix, 14, 19

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1075096.1075117. URL http://dx.doi.org/10.3115/1075096.1075117. 28

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29:19–51, March 2003. 19

Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417–449, December 2004. 19, 27

Franz Josef Och, Christoph Tillmann, Hermann Ney, and Lehrstuhl Fiir Informatik. Improved alignment models for statistical machine translation. In *University of Maryland, College Park, MD*, pages 20–28, 1999a. 30

Franz Josef Och, Christoph Tillmann, Hermann Ney, and Lehrstuhl Fiir Informatik. Improved alignment models for statistical machine translation. In *University of Maryland, College Park, MD*, pages 20–28, 1999b. 21

Franz Josef Och, Nicola Ueffing, and Hermann Ney. An efficient a* search algorithm for statistical machine translation. In *In Data-Driven Machine Translation Workshop*, pages 55–62, 2001. 27

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. Syntax for statistical machine translation. In *Tech. Rep. Summer Workshop Final Report*, Johns Hopkins University, Baltimore, USA, 2003. 22

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 29, 30

Aaron B. Phillips and Ralf D. Brown. Training machine translation with a second-order taylor approximation of weighted translation instances. In *Machine Translation Summit XIII*, 2011. 37

MJD Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 1964. doi: 10.1093/comjnl/7.2.155. 28

M.J.D. Powell. A direct search optimization method that models the objective and constraint functions by linar interpolation. In *In Advances in Optimization and Numerical Analysis, Proceedings of the sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico, volume 275*, pages 51–67. Kluwer Academic Publishers, 1994. 48

William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2002. ISBN 0521750334. 28

Brian Roark, Murat Saraclar, and Michael Collins. Discriminative n-gram language modeling. *Comput. Speech Lang.*, 21, April 2007. 25

Germán Sanchis-Trilles and Francisco Casacuberta. Bayesian adaptation for statistical machine translation. In *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*, SSPR&#38;SPR'10, pages 620–629, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-14979-0, 978-3-642-14979-5. URL `http://dl.acm.org/citation.cfm?id=1887003.1887075`. 37

Holger Schwenk. Continuous space language models. *Computer Speech and Language*, 21:492–518, July 2007. 27

Holger Schwenk. Investigations on large- scale lightly-supervised training for statistical machine translation. In *IWSLT*, pages 182–189, 2008. 34

Holger Schwenk and Jean Senellart. Translation model adaptation for an Arabic/French news translation system by lightly-supervised training. In *MT Summit*, 2009. 4, 34

Holger Schwenk, Daniel Déchelotte, and Jean-Luc Gauvain. Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 723–730, 2006. 27

Holger Schwenk, Patrik Lambert, Loïc Barrault, Christophe Servan, Sadaf Abdul-Rauf, Haithem Afli, and Kashif Shah. Lium's smt machine translation systems for wmt 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 464–469, Edinburgh, Scotland, 2011. Association for Computational Linguistics. 65

Rico Sennrich. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549, Avignon, France, 2012. Association for Computational Linguistics. 37

Kristie Seymore and Ronald Rosenfeld. Using story topics for language model adaptation, 1997. 32

Kashif Shah, Loïc Barrault, and Holger Schwenk. Translation model adaptation by resampling. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 392–399, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-71-8. URL http://portal.acm.org/citation.cfm?id=1868850.1868909. 6

Kashif Shah, Loïc Barrault, and Holger Schwenk. Parametric weighting of parallel data for statistical machine translation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1323–1331, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1148. 6, 73

Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. Translating with non-contiguous phrases. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 755–762. Association for Computational Linguistics, 2005. 16

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 223–231, 2006. 30

Matthew Snover, Bonnie Dorr, and Richard Schwartz. Language and translation model adaptation using comparalble corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 857–866, 2008a. 35

Matthew Snover, Bonnie Dorr, and Richard Schwartz. Language and translation model adaptation using comparable corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 857–866, Honolulu, Hawaii, October 2008b. Association for Computational Linguistics. 35

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 259–268, Athens, Greece, March 2009. Association for Computational Linguistics. 30

Harold Somers. Review article: Example-based machine translation. *Machine Translation*, 14:113–157, 1999. 13

Fei Song and Bruce W. Croft. A general language model for information retrieval. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321, New York, NY, USA, 1999. ACM Press. 25

Andreas Stolcke. Srilm—an extensible language modeling toolkit. *In Proceesings of the 7th International Conference on Spoken Language Processing (ICSLP 2002*, pages 901–904, 2002. doi: 10.1.1.157.2429. 59

Christoph Tillmann and Hermann Ney. Word re-ordering and dp-based search in statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics - Volume 2*, COLING '00, pages 850–856, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/992730.992769. URL http://dx.doi.org/10.3115/992730.992769. 27

Nicola Ueffing. Using monolingual sourcce language data to improve MT performance. In *IWSLT*, pages 174–181, 2006. 34

Nicola Ueffing. Transductive learning for statistical machine translation. In *Association for Computational Linguistics*, pages 25–32, 2007. 34

Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, pages 836–841, Morristown, NJ, USA, 1996. Association for Computational Linguistics. 19

Ye-Yi Wang and Alex Waibel. Modeling with structures in statistical machine translation. In *Proceedings of the 17th international conference on Computational linguistics - Volume 2*, COLING '98, pages 1357–1363, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics. doi: 10.3115/980432.980790. URL http://dx.doi.org/10.3115/980432.980790. 27

John S. White. The arpa mt evaluation methodologies: Evolution, lessons, and further approaches. In *Proceedings of the 1994 Conference of the Association for Machine Translation in the Americas*, pages 193–205, 1994. 29

Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23:377–403, September 1997. 22

Dekai Wu, , Dekai Wu, and Hongsing Wong. Machine translation with a stochastic grammatical channel. In *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL*, pages 1408–1414, 1998. 27

Hua Wu, Haifeng Wang, and Chengqing Zong. Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 993–1000, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-905593-44-6. URL http://dl.acm.org/citation.cfm?id=1599081.1599206. 33

Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530. Association for Computational Linguistics, 2001. 22

Chong Ho Yu. Resampling methods: Concepts, applications, and justification. In *Practical Assessment Research and Evaluation*, 2003. 45

Richard Zens, Franz Josef Och, and Hermann Ney. Phrase-based statistical machine translation. In *German Conference on Artificial Intelligence*, pages 18–32, 2002. 21

Bing Zhao, Matthias Ech, and Stephen Vogal. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004a. 33, 36

Bing Zhao, Stephan Vogel, Matthias Eck, and Alex Waibel. Phrase pair rescoring with term weighting for statistical machine translatio. In *EMNLP*, pages 206–213, 2004b. 37