Improving Statistical Machine Translation Through Adaptation And Learning

Carlos A. Henríquez Q.

Thesis advisor:

Prof. Dr. José B. Mariño Acebal

Thesis co-advisor:

Rafael E. Banchs.

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Thesis presented to obtain the qualification of Doctor from the Universitat Politècnica de Catalunya

Signal Theory and Communications Ph.D. Program

TALP Research Center, Speech Processing Group

Department of Signal Theory and Communications

Universitat Politècnica de Catalunya

Barcelona, January 2014

Abstract

This thesis proposes a new method to improve a Statistical Machine Translation (SMT) system using post-editions of translation outputs. The strategy can be related with domain adaptation, where the in-domain data correspond to post-editions coming from real users of the SMT system. The method compares the post-editions with the translation output in order to automatically detect where the decoder made a mistake and learn from it. Once the errors have been detected, a word alignment is computed between input and post-edition to extract translation units that are then incorporated into the baseline system to fix those errors for future translations. Results show statistically significant improvements with a post-edited collection that is only 0.5% the size of the training material. A qualitative analysis is also studied to validate this results. Improvements are mostly lexical and of word reordering, followed by morphological corrections. The strategy, which introduces the concepts of Augmented Corpus, similarity function and Derived Units, is tested with two SMT paradigms (N-gram-based and Phrase-based), two language pairs (Catalan-Spanish and English-Spanish) and in different domain adaptation scenarios, including an open world domain where the system was adapted to request of any domain collected from real users over the internet, all giving similar results. The results obtained are part of project FAUST (Feedback Analysis for User adaptive Statistical Translation), a project from the Seventh Framework Programme of the European Commission.

Keywords:

statistical machine translation, domain adaptation, feedback, translation model, linear combination, similarity function, augmented corpus, derived units

Resumen

Esta tesis propone un nuevo método para mejorar un sistema de Traducción Automática Estadística (SMT por sus siglas en inglés) utilizando post-ediciones de sus traducciones automáticas. La estrategia puede asociarse con la adaptación de dominio, considerando las post-ediciones obtenidas a través de usuarios reales del sistema de traducción como el material del dominio a adaptar. El método compara las post-ediciones con las traducciones automáticas con la finalidad de detectar automáticamente los lugares en los que el traductor cometió algún error, para poder aprender de ello. Una vez los errores han sido detectados se realiza un alineado a nivel de palabras entre las oraciones originales y las post-ediciones, para extraer unidades de traducción que son luego incorporadas al sistema base de manera que se corrijan los errores en futuras traducciones. Nuestros resultados muestran mejoras estadísticamente significativas a partir de un conjunto de datos que representa en tamaño un $0,5\%$ del material utilizado durante el entrenamiento. Junto con las medidas automáticas de calidad, también presentamos un análisis cualitativo del sistema para validar los resultados. Las mejoras en la traducción se observan en su mayoría en el léxico y el reordenamiento de palabras, seguido de correcciones morfológicas. La estrategia, que introduce los conceptos de corpus aumentado, función de similaridad y unidades de traducción derivadas, es probada con dos paradigmas de SMT (traducción basada en N-gramas y en frases), con dos pares de lengua (Catalán-Español e Inglés-Español) y en diferentes escenarios de adaptación de dominio, incluyendo un dominio abierto en el cual el sistema fue adaptado a través de peticiones recogidas por usuarios reales a través de internet, obteniendo resultados similares durante todas las pruebas. Los resultados de esta investigación forman parte del projecto FAUST (en inglés, *Feedback Analysis for User adaptive Statistical Translation*), un proyecto del Séptimo Programa Marco de la Comisión Europea.

Palabras claves:

traducción automática estadística, adaptación de dominio, retroalimentación, modelo de traducción, combinación lineal, función de similitud, corpus aumentado, unidades derivadas

Acknowledgements

Although only one author appears in the title of this thesis, it would have not been possible without the collaboration and support of many people. This page is to thank you all.

First of all, I like to thank my advisors José B. Mariño and Rafael E. Banchs. You have guided me through these years, offering your unconditional advice, patience and time. You are an excellent team that combines reflection and detail in the small things and share the same love for research and knowledge. I also thank you Rafael for your effort in helping me coming to Barcelona to begin my graduate studies and again for the support and friendship you and your family offered me during my stay in Singapore.

I also thank you, Adolfo, Marta and Lluis, for your support and time and for letting me be part of your team. Most of my experiments and ideas were the result of working in collaboration with you. A special mention to Marta, thank you for trusting me and looking after me, and for sending me your support from all corners of the world.

Speaking of different parts of the world, I must also thank my parents. You have taught me the importance of giving the best of me everyday and have giving me your support and love in all my decisions.

I am immensively grateful to Cristina. You are always there to give me support and help me stay on track and you always encourage me when I need it. Your company, friendship and love have made this thesis possible.

My graduate studies came together with the experience of moving to a new country, far from home and friends. This personal adventure would have been very diffilcult to bear if it were not for the friendship of Alan. I am grateful for all the experiences we have shared together: travels, conversations, discussions and study sessions.

Finally, apart from the people who have being there for me these years, I also thank the TSC Department and the TALP group at the UPC, for the opportunity of working with them, the Spanish Ministry of Economy and Competitiveness for giving me a FPI grant to support my research, Dr. Li Haizhou and the Institute for Infocomm Research for the opportunity of doing a short term stay in Singapore and the FAUST project for also supporting the research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the last decades, cultures and economies from a wide range of countries have been mixing and learning from each other through a process known as globalization. What started mainly as an economic concept with specific examples from multi-national corporation fusions, free commerce and other treaties has spread to scientific collaboration, news dissemination and a daily basis human interaction.

Nowadays it is possible and relatively easy to find information in any language. The human need for information has made people deal with these contents and to find a way to understand them. It could be by learning the new language or by translating the information source. Machine translation (MT) is an automatic way to solve the latter.

Although MT ideas have been studied since 1955 [83], state of the art systems still cannot achieve perfect translations and is not uncommon to see some morphological or semantic errors in their outputs. Nevertheless, their use has become more accessible now that big companies like Microsoft[1] and Google[2] have made available free web versions of their MT systems to all Internet users.

On the other hand, social networks like Facebook and Linkedin and the microblogging site Twitter have encouraged users to share information, teach each other and suggest things to do every day. If we add this social behavior to our translation systems, we would have users that are able to suggest improvements, better translation, and general ideas that could make the systems better.

This is not something that companies have not seen, though. Most of the translation systems you can find on-line, provide a text area for users to suggest a better translation (like Google translator) or a ranking system for them to use (like Microsoft's). Facebook also encourage user's feedback. Sometimes when you see new content in your language, you have the possibility to decide if you like the translation or not, and if you find content in a different language in your timeline, Facebook offers you the possibility

[1] http://www.bing.com/translator

[2] http://translate.google.com/

of translating it. It is worth mentioning that the translation services provided by Facebook are also host by Microsoft.

The problem is that even though MT systems can perform with certain quality in a close domain, like tourist information or weather forecast, its performance is not easily extrapolated to other domains than the one they are built for. For that reason, MT adaptation to specific domains or user feedback is still an active research area.

The European Union (EU) is particularly interested in MT. Being an organization with 24 official languages, it is concerned about finding strategies which make easy to share information between their members. According to European Parliament (EP) statistics[3] multilingualism expenditure represents over one third of the total expenditure of Parliament and the EU system on average requires over 2000 translators and 80 interpreters per day. Just in the first half of 2007, the EP translated 673,000 pages. With tools like MT systems, translators do not need to start from scratch every time they have to translate a document.

Moreover, MT systems have proved to be useful in emergency situations like Haiti's earthquake on January 2010, where they helped volunteers increase their productivity and their capacity to help [58].

MT systems have helped us break barriers. It let us read news from other languages and communicate with different people, it helps us learn new languages and, just like it did 2010, it allows the world to faced together situations that otherwise would have been even more difficult to solve.

1.1 Problem Statement

With the arrival of free on-line MT, came the possibility to improve those systems with the help of their daily users. One of the methods to achieve such improvements is to ask to users themselves for a better translation. It is possible that the system had made a mistake and if the user is able to detect it, it would be a valuable help to let the user teach the system where it made the mistake so it does not make it again if it finds a similar situation.

In 2009, as part of the Seventh Framework Programme of the European Commission [4], the FAUST project started with the goal of developing "machine translation (MT) systems which respond rapidly and intelligently to user feedback" [5]. Specifically, one of the project objective was to "develop mechanisms for instantaneously incorporating user feedback into the MT engines that are used in production environments, ...". As a member of the FAUST project, this thesis focuses on developing one such mechanism.

[3] http://www.europarl.europa.eu/sides/getDoc.do?language=EN&reference=20071017FCS11816

[4] http://cordis.europa.eu/fp7

[5] http://www.faust-fp7.eu/faust/Main/WebHome

1.2 Objectives

There are different approaches of MT systems. The contributions in this thesis are focused in a specific approach known as Statistical Machine Translation systems (SMT), which based on data collected from translated documents and their originals.

Formally, the general objective of this work is to design and implement a strategy to improve the translation quality of an already trained SMT system, using translations of input sentences that are corrections of the system's attempt to translate them.

To address this problem we divided it in three specific objectives:

1. Define a relation between the words of a correction sentence and the words in the system's translation, in order to detect the errors that the former is aiming to solve.
2. Include the error corrections in the original system, so it learns how to solve them in case a similar situation occurs.
3. Test the strategy in different scenarios and with different data, in order to validate the applications of the proposed methodology.

1.2.1 Thesis outline and main contributions

This disseration is organized as follows:

Chapter 2 gives an overview of the state of the art in SMT, it covers the different approaches for SMT, the evaluation metrics and the specific research areas this thesis is focused on: domain adaptation, post-editions and user feedback.

Chapter 3 presents the method proposed to address the first objective, defining a relation between a correction sentence and a system's translation. The method is called Translation-based Word Alignment and introduces the concept of an Augmented Corpus.

Chapter 4 focuses on the second and third objectives. It first describes how to integrate the information collected from the correction sentences into the original MT system and then it explores the technique in different scenarios of domain adaptation. The experiments started with two very close languages as Catalan-Spanish; the next experiment switched to English-Spanish in a domain adaptation task, using in-domain target data as controlled post-editions; we then studied the effects of using real post-editions provided by research experts in the field; finally, our last experiments give results in a open enviroment, where the system was available to the public through a web interface, there was no specific domain to adapt to and the feedback was provided freely be internet users.

Chapter 5 highlights the contributions made by this thesis and propose extensions to this research.

The thesis ends with **Appendix A** describing the corpora used for the different experiments and **Appendix B** listing the publications made during this research.

The main contributions of this thesis are the following ones:

- We defined a similarity function that compares an MT system output with a translation reference for that output and align the errors made by the system with the correct translations found in the reference. This information is then used to compute an alignment between the original input sentence and the reference.
- We defined a method to perform domain adaptation based on the alignment mentioned before. Using this alignment with an in-domain parallel corpus, we extract new translation units that correspond both to units found in the system and were correctly chosen during translation and new units that include the correct translations found in the reference. These new units are then scored and combined with the units in the original system in order to improve its quality in terms of both human an automatic metrics.
- We succesfully applied the method in a new task: to improve a SMT translation quality using post-editions provided by real users of the system. In this case, the alignment was computed over a parallel corpus build with post-editions, extracting translation units that correspond both to units found in the system and were correctly chosen during translation and new units that include the corrections found in the feedback provided.
- The method proposed in this dissertation is able to achieve significant improvements in translation quality with a small learning material, corresponding to a 0.5% of the training material used to build the original system.

Chapter 2

State of the Art

This chapter provides a brief overview of the current trends of Statistical Machine Translation (SMT). It begins with an overview of machine translation and illustrating in a high level what is the idea behind SMT. Once it stablishes the motivation, the chapter continues with a more detailed description of two different ways for building an SMT system, including the main tools available to do so. Building a machine translation system also involves defining metrics to determine whether the translations are good or not. Therefore, after describing how to build an SMT system, we focus on the evaluation metrics that are used to measure its quality. Finally, the chapter ends explaining Domain Adaptation and User Feedback, the main research areas of this Ph.D. thesis.

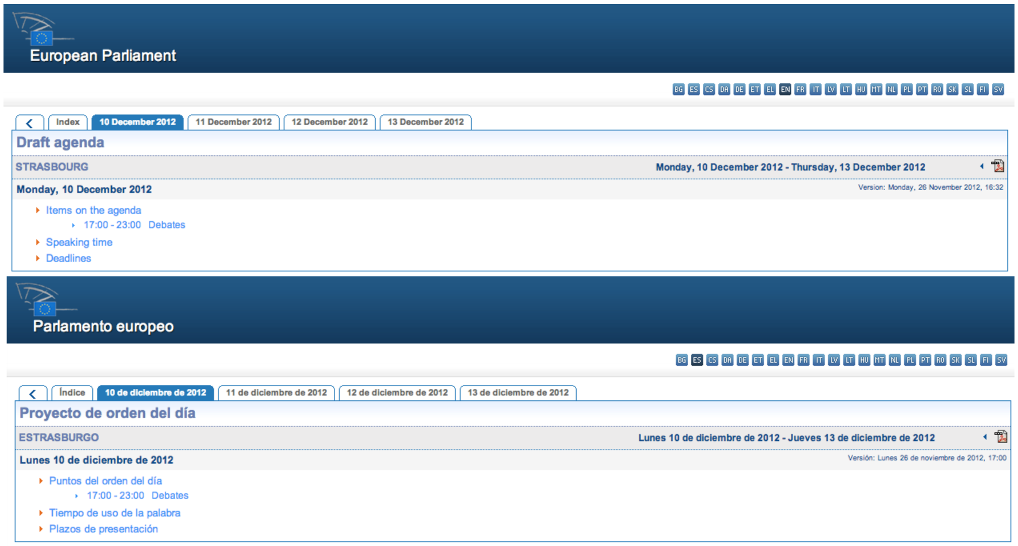
2.1 Different Approaches To Machine Translation

Machine Translation (MT) is the area of computational linguistics which studies systems that are able to translate texts from one human language into another. Depending on the chosen strategy to build translations, systems can be categorized as follows:

Rule-based systems: MT systems in this category rely on linguistic rules (usually designed by human experts) to describe the translation process.

Data-driven systems: This category includes all the systems that use parallel corpora and previously translated documents to produce new translations.

Statistical Machine Translation (SMT) belongs to the second category. Besides SMT systems, we could also find in the Data-driven category two other approaches: Translation Memory and Example-based. Both approaches generate translations by extracting and matching pieces from previously translated corpora and joining them to create new ones. Its difference is that the extraction phase is performed



Draft agenda

STRASBOURG Monday, 10 December 2012 - Thursday, 13 December 2012

Monday, 10 December 2012 Version: Monday, 26 November 2012, 16:32

- Items on the agenda
    - 17:00 - 23:00 Debates
- Speaking time
- Deadlines

Proyecto de orden del día

ESTRASBURGO Lunes 10 de diciembre de 2012 - Jueves 13 de diciembre de 2012

Lunes 10 de diciembre de 2012 Versión: Lunes 26 de noviembre de 2012, 17:00

- Puntos del orden del día
    - 17:00 - 23:00 Debates
- Tiempo de uso de la palabra
- Plazos de presentación

Figure 2.1: English and Spanish version of the Europarl web page showing the draft agenda of a future session. In the top right side you are able to choose between 22 languages.

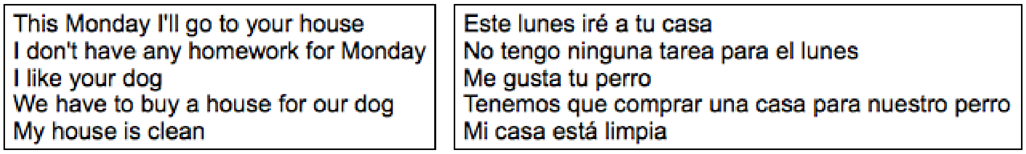
manually on the Translation Memory approach and automatically on Example-based systems. An overview of Machine Translation history and references to their different approaches can be found in Koehn's book "Statistical Machine Translation" [53].

2.2 A first look at Statistical Machine Translation

The first step to build a SMT system is gathering a Parallel Bilingual Corpus. A parallel bilingual corpus (hereafter, a parallel corpus) is a collection of documents in two languages (the source and the target) divided in sentences, where each sentence in the source has a corresponding translation sentence in the target. Good examples of parallel corpora are the European Parliament (Europarl) sessions [52]. By law, all sessions in the European Parliament must be translated in all the official languages. Figure 2.1 shows the web page of the Europarl with the draft of a session both in English and Spanish.

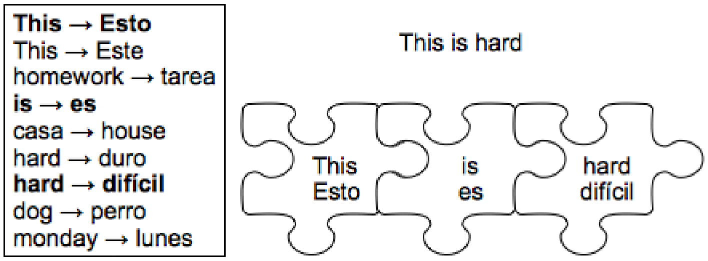
Once we have a parallel corpus, the next step to build an SMT system is to align it at the word level, and here is where the "S" in SMT begins to play an important role.

Suppose for a moment that we have a small parallel corpus and we start reading its content. Comparing line by line from the source and target sides, we can begin drawing some conclusions about the meaning of the most frequent words. Figure 2.2 shows five sentences in English and their corresponding translation in Spanish. Even if we know no Spanish we can make an educated guess that the translation of *Monday*, *house* and *dog* are *lunes*, *casa* and *perro*, respectively, because every time we see the English word on the left side, we see the Spanish one on the right side. Following the same idea, a computer can compare millions of sentences several times and based on statistics it can go from "educated guess" to statistical word matching.



This Monday I'll go to your house	Este lunes iré a tu casa
I don't have any homework for Monday	No tengo ninguna tarea para el lunes
I like your dog	Me gusta tu perro
We have to buy a house for our dog	Tenemos que comprar una casa para nuestro perro
My house is clean	Mi casa está limpia

Figure 2.2: A parallel corpus of five sentence in English and Spanish



This → Esto
This → Este
homework → tarea
is → es
casa → house
hard → duro
hard → difícil
dog → perro
monday → lunes

Figure 2.3: An example of phrases from a phrase table on the left and how they can be combine to translate "This is hard". The pieces used for the translation are boldfaced in the table.

The result from this alignment is a "phrase table" built from our parallel corpus. These phrases can be seen as small building blocks. Every time we want to translate a new sentence, we just look among our phrases to find a match in the source side and then we start to combine the phrases until we have a subset that matches all together. Figure 2.3 shows an example of a phrase table and how they are selected to translate a new sentence.

Basically, this is how SMT systems work. We skipped some steps for simplicity because the purpose of this section was just to give you a first glance. The following section will define formally each of the steps mentioned above for the two SMT approaches studied in this thesis: Phrase-based and N-gram based SMT. Appart from these approaches, other strategies to perform machine translation are mentioned in Section 2.6.3.

2.3 A General Framework for SMT

The secret ingredient in Statistical Machine Translation comes from probability theory and the concepts of joint and conditional probability and the Bayes' theorem, first presented in [8]. The joint probability of events A and B is defined as the probability of their intersection, i.e., of the two events happening, formally:

$$P(A \cap B) = P(A)P(B|A) \tag{2.1}$$

where $P(A)$ is the probability of event A and $P(B|A)$ is the conditional probability of event B given that event A happened. In English, the probability for two events to occur is defined as the probability of the first multiplied by the probability of the second, given that the first already happened. Because the intersection of two sets is commutative, we can say that $P(B \cap A) = P(A \cap B)$. Therefore,

$$\begin{aligned} P(B \cap A) &= P(A \cap B) \\ P(B)P(A|B) &= P(A)P(B|A) \\ P(A|B) &= \frac{P(A)P(B|A)}{P(B)} \end{aligned} \tag{2.2}$$

where equation (2.2) is known as the Bayes' theorem.

Statistical Machine Translation relies on the translation of a source language sentence f (usually referred to as "French") into a target language sentence $\hat{e}$ (usually referred to as "English"). Among all possible target language sentences e we choose the one with the highest probability, as show in equation (2.4):

$$\hat{e} = \arg\max_{e} \left[P\left(e|f\right) \right] \tag{2.3}$$

$$= \arg\max_{e} \left[P\left(e\right) P\left(f|e\right) \right] \tag{2.4}$$

where $P(e)$ is the probability that the target language model gives to the sentence e (for sake of simplicity, we call $P(e)$ the target language model) and $P(f|e)$ is known as the translation model. This probability decomposition based on Bayes' theorem is known as the source-channel approach to statistical machine translation [12]. We can see that in this decomposition, the denominator from equation (2.2) has dissapeared. This is a safe assumption as the probability $P(f)$ is a positive constant for all hypotheses e and therefore it can be discarded. The source channel approach allows to model independently the target language model, and the translation model, which are the two key concepts in SMT.

The translation model estimates how well different words in the foreign language are translations of words in the source language. You can find the details of the different translation models considered in this thesis in Section 2.6.1 and 2.6.2 for Phrase-based SMT and N-gram based SMT, respectively. The language model, on the other hand, measures how good a given hypothesis is written in the target language, i.e., the fluency of hypothesis e. A more complete description on the language model can be found in Section 2.4. The search process is represented as the argmax operation and it is briefly covered in Section 2.8.

Later on, a variation was proposed by Och and Ney in 2003 [67] based on a log-linear model approach that first appeared in [9]. It generalizes the probabilities presented in (2.4) to allow for using more than two models and to weight them independently as can be seen in equation (2.5):

$$\hat{e} = \arg\max \left[\sum_{m=1}^{M} \lambda_m h_m(f,e) \right] \tag{2.5}$$

How we compute those weights for each model is discussed in Secion 2.9. Notice that if we set $M = 2$,$\lambda_m = 1$ and if we consider h_1 and h_2 as the logarithm of the probality given by the target language model and the source translation model we would be back to equation (2.4). This framework allow us to consider additional features that model different characteristics of our translation hypotheses e. The most common features included in (2.5) are:

Sentence length model: Usually called word penalty, it compensates the system tendency to prefer shorter translations which is due to the target language model. It is defined as a function depending on the number of words contained in a translation hypothesis.

Lexical models: Additional source-to-target and target-to-source models are added to the set of features in order to improve the confidence on a translation hypothesis. They could use word-to-word IBM model 1 probabilities (discussed in Section 2.5) to define them or just a maximum likelihood lexical table obtained from an aligned corpus.

Reordering model: It models how words move around from their positions within the source language sentence to their final positions in their translations. Without it, some sentences could only be translated correctly if both languages had a similar word order. The most common reordering model is described in Section 2.7.

The following sections will describe the different features considered in equation (2.5). We start with the language model and the alignment process, as they are common to the different SMT approaches. Then, we describe the translation model for different SMT approaches: Phrase-based, N-gram based and Hierarchical Phrase-based. We continue with the definition of reordering model and then we briefly describe the search process or decoding (the one that actually computes the translation) and the optimizaction process, which is the one that sets the λ values for the different features.

2.4 The Language Model

As mentioned earlier, a language model measures how good a sentence is written in a language. State-of-the-art SMT systems commonly use language models based on probabilities extracted from a monolingual training corpus. Theoretically, given a sentence S formed by words s_1, s_2, ..., s_k the language model will compute the probability of S as:

$$\begin{aligned}
P(S) &= P(s_1 \cap s_2 \cap \ldots \cap s_k) \\
&= P(s_k \cap s_{k-1} \cap \cdots \cap s_1) \\
&= P(s_k|s_{k-1},\ldots,s_1)P(s_{k-1}|s_{k-2},\ldots,s_1)\cdots P(s_2|s_1)P(s_1) \\
P(S) &= \prod_{i=1}^{k} P(s_i|s_{i-1},\ldots,s_1)
\end{aligned}$$

In practice, this computation is unfeasible as k grows larger, hence an assumption is made that the probability of a word s_i only depends on the previous $n-1$ words and not on all the history before it.

$$\begin{aligned}
P(S) &= \prod_{i=1}^{k} P(s_i|s_{i-1},\ldots,s_1) \\
&= \prod_{i=1}^{k} P(s_i|s_{i-1},\ldots,s_{i-(n-1)})
\end{aligned}$$

The assumption of looking at only n words instead of the whole sentence is the reason why this type of language model is called an n-gram model. Common values for n in state-of-the-art SMT are 3 and 5 [48]. The idea behind them is to divide the sentence into pieces that are small enough to be frequent but large enough to contain some language information [47].

In order to calculate the conditional probability of an n-gram, the model bases its computation in the the maximum likelihood. For instance, given $n = 3$:

$$P(s_k|s_{k-1},s_{k-2}) = \frac{count(s_k,s_{k-1},s_{k-2})}{count(s_{k-1},s_{k-2})} \tag{2.6}$$

where $count(s_k,s_{k-1},s_{k-2})$ is the number of times the n-gram (s_k,s_{k-1},s_{k-2}) appeared in the training corpus.

However, the final probability of a n-gram is not directly derived from equation (2.6), because it does not provide good probability estimates for unseen n-grams. To overcome this problem several smoothing algorithms have been proposed like Good-Turing [40] or Kneser-Ney [49], which take a portion of the probability mass of the model and assign it to a new n-gram that represents any unseen n-gram.

In the SMT research community, the two most commonly used software libraries for computing n-gram based language models are SRILM [75] and IRSTLM [32].

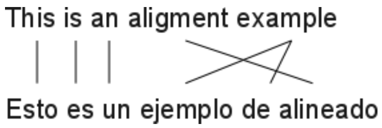


Figure 2.4: An alignment example.

2.5 The Aligning Process

The first step to build a translation model is to align a parallel corpus at the word level. An alignment is a many-to-many relation between the words of a source language sentence and its corresponding translation in the target language. An alignment example can be seen in Figure 2.4.

In order to obtain such a relation between the sentences, most of the state-of-the-art SMT systems use the IBM approach, proposed by Brown et. al. [13]. The idea is to estimate a translation model with the word alignments as a hidden variable, as shown in equation (2.7):

$$p\left(f|e\right) = \sum_{a_i} p\left(f, a_i|e\right) \tag{2.7}$$

where the term $p(f, a_i|e)$ is defined as the multiplication of simpler models:

$$p(f, a_i|e) = \prod n(\phi|e) \prod t(f|e) \prod d(\pi|\tau, \phi, \epsilon) \tag{2.8}$$

This approach uses an Expectation-Maximization (EM) algorithm to train the fundamental models used to estimate the probabilities in equations (2.8) and (2.7) [13]. These models are:

- Fertility model $n\left(\phi|e\right)$: it accounts for the probability that a target word e generates ϕ source words in the source sentence.
- Lexicon model $t\left(f|e\right)$: it accounts for the probability that source word f is a translation of target word e.
- Distortion model $d\left(\pi|\tau, \phi, \varepsilon\right)$: it explains the phenomenon of placing a source word in position π given the target word is placed in position τ, from a target sentence of ϕ words and a source sentence of ε words.

Notice that our final objective, as stated in Section 2.3, is to find the English sentence $\hat{e}$ with the highest probability, and because of equation (2.4), we must focus on the backward probability $p(f|e)$.

In 1999, Kevin Knight [50] explained the IBM approach as a generative process, or a sequence of steps to perform the translation modeled by this backward probability: first, the target word e multiplies itself to get k copies $e_1, \ldots, e_k$ (the fertility model), then the resulting copies translate into words in the source language (the lexicon model) and last, the words move around to their final positions in the source sentence (the distortion model). This process is known as the IBM Model 3.

In order to compute a good estimation for $p(f|e)$, the complete IBM approach introduces two other models apart from IBM Model 3 that are ran in previous steps to initialize the IBM Model 3 EM algorithm. These models are unsurprisingly called IBM Model 1 and IBM Model 2 and each of them are computed with a EM algorithm as well. Their purposes is to simplify equation (2.8) so it can be built incrementaly with each IBM Model.

- IBM Model 1 simplifies equation (2.8) to consider only the lexicon model $p(f, a_i|e) = \prod t(f|e)$. It is the first model ran and its Viterbi [80] alignment is used to initialized the lexicon model in the next step, IBM Model 2.

- IBM Model 2 involves word translations and adds the distortion model into the picture. Its objective is to penalize alignments where the distance between the linked words is very long, even if they are good translation of each other. The distortion model proposed in this model differs from the one in equation (2.8), here the model considers the reverse distortion probability $\hat{d}(\tau|\pi, \phi, \epsilon)$. Its Viterbi alignment is transfered to IBM Model 3 so the EM algorithm starts with better values for its lexicon and distortion model.

At the end, a greedy approach computes the final alignment among all posible a_i. Because of the fertility model, there may be links between the same source word and different reference words. Therefore, the result is a one-to-many alignment. In the same way, the process can be ran again swapping the source and target language to obtain a many-to-one alignment. Finally, a heuristic symmetrization algorithm compute the final many-to-many alignment which reduces the effect of incorrectly aligned multi-word units. This symmetrization algorithm could be an intersection of alignments, a union or the grow-diag-final-and heuristic [54].

From the final many-to-many alignment, the probabilities that define the translation model are extracted and also some of the additional features, like the lexical models (some systems directly use the IBM Model 1 as the final lexical model).

Current SMT systems use an open-source tool called GIZA++ [67], which implements the IBM models and replaces IBM Model 2 with the Hidden Markov Model proposed in [81], to compute the one-to-many and the many-to-one most probable alignment. Another tools that have been used recently are MGIZA++ [38], a multi thread alignment tool based on GIZA++ and the Berkeley aligner [60] which is also based on the IBM models.

2.6 The Translation Model

As it was mentioned before, the translation model estimates how well different words in the foreign language are translations of words in the source language. Different approaches have been proposed to model such estimations. The following sections describe three valid alternatives of translation models.

2.6.1 Phrase-based Translation

This approach made the leap from word-to-word translation (as it was defined by the IBM models on Section 2.4) to translations based in phrases. A phrase is nothing more than a sequence of words (without linguistic motivation). Following the same idea, a bilingual phrase is defined as a pair of source and target phrases that are consistent with the word alignment [87] (there are no links between a word in a bilingual phrase and a word outside of it). Example of phrases that can be extracted from the aligned sentences in Figure 2.4 are $\langle This\ is, Esto\ es \rangle$, $\langle alignment\ example, ejemplo\ de\ alineado \rangle$ and $\langle This\ is\ an, Esto\ es\ un \rangle$, among others. An example of something that is not a phrase in Figure 2.4 is $\langle an\ alignment\ example, ejemplo\ de\ alineado \rangle$ because the word "an" is aligned with the Spanish word "un" which is outside the bilingual phrase.

Phrase-based systems are able to learn many bilingual correspondences that word-to-word translation systems cannot. For instance, in a word-to-word English-Spanish SMT system the phrasal verb "to look after" cannot be pair with "cuidar" as a unit; another example could be "taxi driver" which separated translates to "taxi" and "chofer" but that has a specific Spanish word for it, "taxista".

Translation probabilities for phrases are estimated as relative frequencies over all bilingual phrases in the corpus. Therefore, the translation model is built first by extracting all possible bilingual phrases for each parallel sentence in corpus, and then for each phrase (f, e), where f is the monolingual phrase on the source corpus and e the monolingual phrase on the target corpus, two probabilities are computed:

$$P\left(f|e\right) = \frac{N\left(f,e\right)}{N\left(e\right)} \tag{2.9}$$

$$P\left(e|f\right) = \frac{N\left(f,e\right)}{N\left(f\right)} \tag{2.10}$$

where $N\left(f,e\right)$ counts the number of times the phrase f is translated as e and $N\left(f\right)$ and $N\left(e\right)$ the number of times the phrase in the source or the target language correspondingly appears in the training corpus. Logarithms of equations (2.9) and (2.10) are then considered two different features in equation (2.5). Besides the translation probabilites computed in equations (2.9) and (2.10), the translation model in Phrase-based translation also includes two more features for each phrase, which are referred to as lexical weights because they come from the lexical model concept described in Section 2.5.

Formally, having a bilingual phrase (f, e) where f is a source phrase with words $f_1, \ldots, f_n$ and e is a target phrase with words $e_1, \ldots, e_m$ and a word based model $t(f_i|e_j)$ that returns the probability of word f_i given word e_j, the lexical weight $lex(f|e)$ is defined as:

$$lex(f|e) = \prod_{i=1}^{n} \sum_{j=1}^{m} t(f_i|e_j) \tag{2.11}$$

where the lexicon model $t(f_i|e_j)$ could be the result of IBM Model 1 or a word based model derived from frequency counts of links in the final word alignment of the bilingual corpus. Similarly, the lexical weight $lex(e|f)$ is defined in terms of $t(e_i|f_j)$. Just like the translation probabilities, the logarithms of the lexical weights are included in equation (2.5). Currently, the Moses toolkit [55] is an open-source Statistical Machine Translation system that implements Phrase-based machine translation.

2.6.2 N-gram Based Translation

The N-gram based Machine Translation bases its translation model on tuples [61]. Similar to a phrase, a tuple is a bilingual unit with consecutive words both on the source and target side and is consistent with the word alignment. But, unlike phrases, tuple extraction must produce a unique monotonic segmentation of the sentence pair; and last, a tuple cannot be inside other tuple. Following alignment in Figure 2.4, the tuples that can be extracted are: $\langle This, Esto\rangle$, $\langle es, is\rangle$, $\langle a, un\rangle$ and $\langle alignment\ example, ejemplo\ de\ alineado\rangle$. Phrases like $\langle This\ is, Esto\ es\rangle$ and $\langle is\ a, es\ un\rangle$ are not tuples because smaller tuples can be extracted from them such that a monotonic segmentation of the sentence pair is still produced. On the other hand, phrases like $\langle example, ejemplo\ de\rangle$ and $\langle alignment, alineado\rangle$ are not tuples because they would not allow a monotonic segmentation. Another way to see tuples is as the smallest phrases you can extract from a pair of aligned sentences that produce a unique monotonic segmentation of the pair.

Precisely this unique monotonic segmentation requirement is the key in the tuple concept. It allows us to see the translation model as an n-gram based language model as described in Section 2.4, where the language is composed by tuples instead of words. That way, the context used in the translation model is bilingual and implicitly works as a language model with bilingual context as well. In fact, while the language model is required in Phrase-based and other SMT systems, in N-gram based systems it is considered just an optional additional feature.

This alternative approach to a translation model reformulates the translation probability as:

$$P\left(f, e\right) = \prod_{n=1}^{N} P\left(\left(f, e\right)_n \mid \left(f, e\right)_{n-1}, \ldots, \left(f, e\right)_1\right) \tag{2.12}$$

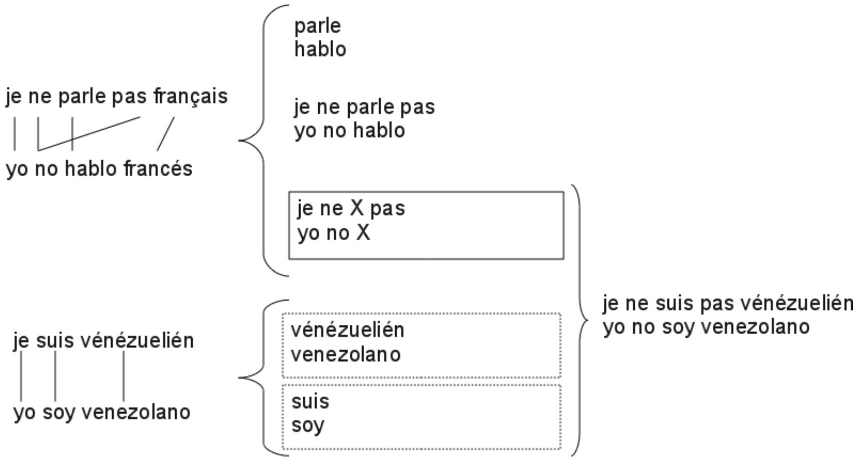


Figure 2.5: The phrase "je ne parle pas | yo no hablo" contains the smaller phrase "parle|hablo", therefore the latter can be extracted to build the hierarchical phrase "je ne X pas | yo no X"

where $(f, e)_n$ is the n-th tuple of hypothesis e for the source sentence f. At the end, like relative frequencies for phrases, the logarithm $\log P(f, e)$ will be another feature of equation (2.5). Lexical weights are usually included as features in addition to the translation model. In this case, the lexical weights are defined per tuple instead of per phrase. Just like Moses for Phrase-based, there is an open source decoder for N-gram based MT called MARIE [23].

2.6.3 Hierarchical Phrase-based Translation

Appart from Phrase-based and N-gram based SMT, another approach to machine translation based in statistics is the Hierarchical Phrase-based Machine Translation. This approach, proposed in 2005 by Chiang et. al. [17, 18], was born as an extension of the Phrase-based approach and it introduced the concept of hierarchical phrases. A hierarchical phrase consists of a phrase that may contain sub-phrases in it. These hierarchical phrase pairs are formally productions of a synchronous context-free grammar which is used to perform the translation as a parsing procedure. This grammar is induced from parallel text without relying on any linguistic annotation or assumption.

The way to build them is by first extracting all regular phrases from the parallel corpus, then a rule for each phrase is defined with terminal elements; and finally if for a given rule we find that another initial phrase exists inside of it, a new rule is define, replacing that initial phrase for a placeholder (a non-terminal character) for any sub-phrase. An example of hierarchical rules between French and Spanish can be seen in Figure 2.5.

Because the number of rules extracted far exceeds the number of phrases typically found in aligned text, a rule filtering strategy was proposed in 2009 [45] which reduces the rule size while maintaining good translation quality. It is based on the number of non-terminal elements and patterns found in them.

As in Phrase-based models, the features for each rule are the relative frequencies and the additional lexical models. For hierarchical Phrase-based models, JOSHUA [59] is an open source toolkit that extracts the translation grammar, trains the models and performs the translation. In its latest versions, Moses [55] also includes hierarchical Phrase-based machine translation.

2.7 The Reordering Model

Earlier in this chapter, it was mentioned that the main benefit of the log-linear model is that the features considered for machine translation could be extended beyond the language and translation models. This section describes one of these additional models: the reordering model.

The reordering model describes how words in the source language move from their original positions in the source sentence to their final positions in the target sentence once they are translated. The distortion model considered during the aligning process in Section 2.5 is a way of describing this behavior.

Depending on the approach used for the translation model (Phrase-based, N-gram based or Hierarchical Phrase-based), different reordering models have been proposed. The most basic reordering model is called distance-based reordering. This model gives a cost linear to the reordering distance. It is usually included, together with more complex models, as a feature when using Phrase-based translation.

In 2004, Tillman [76] proposed a lexicalized reordering model that considers three different moves a phrase can make related to the previous and following phrase: monotone move, swap move and discontinous move. The model computes for each phrase (f, e) the relative frequencies of each move in both directions, i.e. $r(f|e)$ and $r(e|f)$ for a total of six different weights for the same phrase. This model represents the State-Of-The-Art in Phrase-based SMT and is included in the Moses Toolkit. Figure 2.6 (a) shows an example of the different moves.

For N-gram based SMT, reordering occurs as a two steps process. First, after the aligning process, the source words move around in order to avoid crossed links and force a monotonic segmentation with the maximum amount of tuples. This movement of source side words is called unfolding. Figure 2.6 (b) shows the same alignment for the Tillman examples but with unfolded tuples. The translation model is then trained with the unfolded tuples. At the same time the source words move around, the unfolding technique learns this movements and associates them to the Part-Of-Speech tags of the source words to generalize them. The second part of the reordering model comes just before a new sentence is translated. At this moment, the unfolding rules learnt during training are applied over the source words to align the words in the same way the translation model was computed. A detailed explanation together with experiments applying this technique were presented by Crego and Mariño in 2007 [24]. Later in 2011, Crego et. al. [26] presented a new N-gram based SMT system called Ncode which includes, besides the source side reordering just described and the lexical reordering proposed by Tillman, another reordering model for this approach called Linguistically Informed Bilingual n-gram model [25], which was presented in 2010.

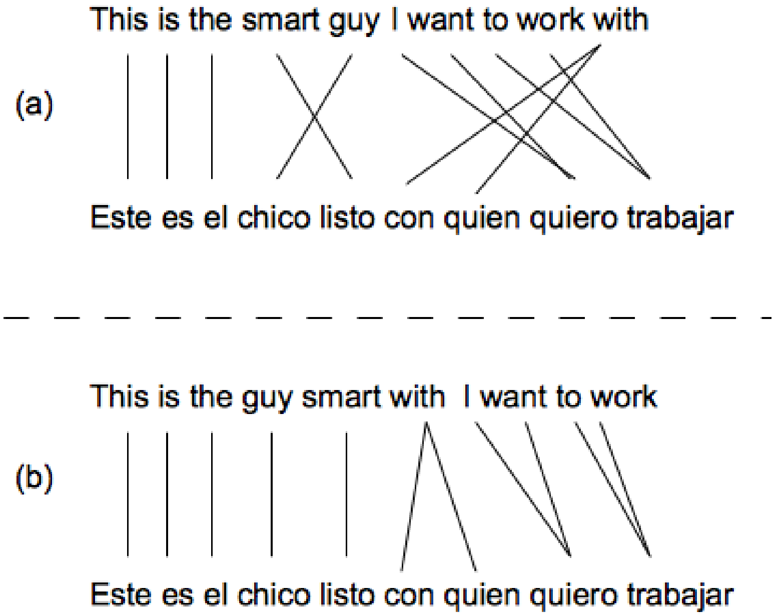

(a)
This is the smart guy I want to work with
Este es el chico listo con quien quiero trabajar
(b)
This is the guy smart with I want to work
Este es el chico listo con quien quiero trabajar

Figure 2.6: (a) Example of moves in lexical reordering:$\langle is, es \rangle$ makes a monotone move respect to both neighbor phrases, $\langle guy, chico \rangle$ makes a swap respect to its previous phrase and $\langle with, con \rangle$ makes a discontinuos move respect to its previous phrase. (b) Example of tuple unfolding for N-gram based translation.

Regarding the Hierarchichal Phrase-based systems, they were originally designed to include possible long reorderings within its translation model thanks to the non-terminal character in their phrases. However, specific reordering models like the one proposed by Galley and Manning [37] have also been designed for this approach to SMT.

2.8 Decoding

Decoding is the process that searches for the best translation hypothesis among "all possible" translations. In the general framework defined in Section 2.3, it is represented by the argmax function of equation (2.5). "All possible" is quoted because in practice, the decoding process does not consider all hypotheses. Only a selected groupd of promising hypotheses is considered according to the scores given by the different features in the log-linear model.

The details of the decoding process vary between SMT approaches. Hierarchical Phrase-based systems perform a CKY parse of the source sentences [18]. N-gram based systems perform a pruned-search of hypotheses using a finite state automata structure and a word lattice as input [26]. Finally, Phrase-based systems apply a beam search algorithm with a A* approach upon considering future costs for hypotheses. A detailed description of the beam search procedure can be found in [53]. Another search algorithm proposed for Phrase-based systems is called "Cube Pruning" [44] and it is described to be faster than the beam search with similar search errors (caused be pruning).

2.9 Tuning or Optimization

The log-linear model presented in Section 2.3 includes a weight λ_m for each feature h_m. The process of setting the λ_m to obtain the best translation is called tuning or optimization.

To perform the tuning process, we need an additional parallel corpus called the development corpus or tuning set. The source side of this corpus is translated using an initial set of values for λ_m and the translation is compared with the target side to measure its quality according to a given metric. A review of evaluation metrics for Machine Translation can be found in Section 2.10. The λ values are then modified according to the optimization strategy used and the process starts over with a new translation to see if the new weights improve the translation quality. The procedure ends when no further improvements are possible or after a determined number of iterations.

In the same way that different translation model, reordering model and decoding strategies have been proposed for SMT, several optimization methods have been designed.

One of the first proposed alternatives followed the downhill simplex strategy [64]. This method, however, is very expensive in terms of computational cost because the system needs to translated the complete development corpus everytime it considers a new set of λ values.

Modern optimization strategies try to avoid this need of many translation working in batches. When translating the development set, a n-best list is generated for each source sentence and an inner loop process starts to update the feature's weights using these lists. Then, the development corpus is translated again using the new weights and the process repeats until the convergence criteria is satisfied. Within this framework of inner loop optimization and outer loop translation, different inner loop algorithms have been proposed:

- Minimun Error Rate Training (MERT): First presented by Och in 2003 [66] and then included in the Moses toolkit by Bertoldi et. al. in 2009 [10] focuses its inner loop in optimizing one weight at a time.

- Pairwise Ranking Optimization (PRO): Proposed by Hopkins and May in 2011 [43] as an alternative of MERT that can handle a high number of features. The proposal considers the optimization as a binary classification problem: the λ values are returned by a linear classifier trained with labeled samples extracted from the n-best list. The label describes whether hypothesis e_i is better than e_j according to the metric chosen.

- Margin Infused Relaxed Algorithm (MIRA): Based on the work from Cherry and Foster [16] it is a batch version of the online algorithm MIRA [19], replacing the translation of each sample with re-ranking of n-best lists.

2.10 Evaluation metrics

Evaluation in machine translation is the process of studying and measuring the quality of a translated text compared to a given reference. Having evaluation metrics is essencial for research, as they tell us how a system is evolving after applying some technique or simply how it performs and how it compares to others. Evaluation metrics can be split in two main categories: human based and automatic metrics.

Human based metrics are quality measures stablished by human evaluators. Human based metrics are the most reliable to assess translation quality, but they tend to be time consuming and expensive to obtain. Among the main human based evaluation metrics we find fluency, adequacy, ranking between systems and more recently, human-targeted translation edit rate.

Automatic metrics are computed without the intervention of human evaluators. They have the benefit of being faster to compute than human evaluation metrics and almost unexpensive. Therefore, they are mostly used during training and development while the human based metrics are used for testing and comparison of final systems.

Having a reference to compare with may bias some systems against others whose hypotheses contains synonyms or paraphrases. For instance, an hypothesis with the term “mobile phone” may be considered a better candidate than other that uses “cell phone”, if in the reference appears the former term, even though both of them have the same meaning.

In order to reduce this bias effect as much as possible, the automatic evaluation metrics compare an hypothesis against multiple references and some of the metrics defined specific strategies that contrarrest these situations. The most used automatic evaluation metrics are BLEU, WER and PER, TER, NIST and METEOR.

2.10.1 Human based evaluation metrics

As mentioned before, human based evaluation metrics are quality measures stablished by humans. The following sections defined fluency, adequacy, Human-targeted Translation Edit Rate, ranking and pair-based comparison.

2.10.1.1 Fluency and Adequacy

Fluency is measured in a discrete scale of 1 to 5. It indicates how natural a translation hypothesis sounds to a native speaker of the target language [47]. For instance, the English sentence “This book is better than the other” is more fluent that “This is better book than the other”, even though they try to express the same meaning.

Similar to fluency, adequacy is also measured in a 1 to 5 discrete scale. Adequacy determines how much of the information contained in the original sentence is preserved in the translation hypothesis. To measure adequacy, the original translation is presented to the human evaluator for comparison (in case the human evaluator is not fluent in the source language, a reference translation is presented instead). Adequacy is about meaning, therefore the sentence “I will call you when I get home” when compared with the reference “I will give you a call when I get home” has a higher adequacy than the alternative “I may call when I get home” because in the first hypothesis the subject is assuring that he will make the call (just like the reference), while in the latter the translation is only expressing a possibility.

2.10.1.2 Ranking between systems and pairwise comparison

Apart from measuring translation quality against references, a system can also be compared with others in order to determine a ranking between them. Comparison between systems are usually performed with two variants: ranking between multiple systems and pairwise comparison, defined in [79]. An application of this two evaluation metrics with real MT systems can be found in [15].

Comparison of systems are also performed using a target reference. In a ranking based evaluation, the human evaluator is presented with hypotheses from different systems (usually between three and five) and he must defined an order between them according to their quality when compared with the reference. A pair based comparison is a decision made between two hypotheses where the evaluator must determine if the former system is better than the latter, if it is the other way around or if it cannot be decided which of the two systems is better.

2.10.2 Automatic evaluation metrics

In contrast to human based evaluation metrics, automatic evaluation metrics have the benefit of being faster to compute and almost unexpensive. They relied on a set of translation references to measure translation quality. In order to facilitate the computation of these automatic metrics, Giménez and Màrquez developed in 2010 Asiya [39] as a contribution to the FAUST project[1], an open source toolkit that offers a set of evaluation metrics and meta-metrics for MT.

The following sections describe the most common automatic metrics used in MT: Word Error Rate, Position independent Error Rate, BLEU, NIST, Translation Edit Rate and METEOR.

2.10.2.1 Word Error rate (WER) and Position independent Error Rate (PER)

The first automatic metrics for machine translation came from the field of speech recognition. WER is defined as the quotient between the minimun number of replacements, insertions and deletions needed

[1] http://www.faust-fp7.eu/faust/

to go from a translation hypothesis to its closest reference and the average length of the references. The numerator in this metric is known as the Levenshtein distance. Formally it is defined as:

$$WER = \frac{\text{\# of edits}}{\text{avg. \# reference words}} \tag{2.13}$$

Position independent error rate [77] is a simpler metric as it does not consider the position of the words in the sentences. It measures the difference in the count of the words occurring in hypothesis and reference divided by the number of words in the reference.

2.10.2.2 BLEU

Before defining the Bilingual Evaluation Understudy [68] (BLEU) metric, we need to introduce the concepts of n-gram precision and *modified* n-gram precision.

The standard n-gram precision of an output given a reference is defined as the number of n-grams in the output which occur in the reference divided by the total number of n-grams in the output. For instance, given the output “This is a close translation” and the reference “This is the best translation”, the 1-gram precision is 3/5, because there are three 1-grams in the output that occur in the reference (*This, is* and *translation*) and there are five 1-grams in total in the output.

The problem with the standard n-gram precision is that you *can* have high precision scores for outputs that are not close to the reference. Let us supposed now that our output is “This This This This This” and we have the same reference as before. In this case our 1-gram precision is 5/5 even though the translation output is worse. That is because unigram “This”, which occurs once in the reference, appears five times in the output, i.e. $count(\text{This}) = count_{output}(\text{This}) = 5$.

The *modified* n-gram precision addresses this problem defining an upper bound for the count of a n-gram m. This upper bound is defined as the maximum number of times the n-gram m appears in the reference, i.e. $count(m) = \min[count_{output}(m), count_{reference}(m)]$. In our two examples, the *modified* 1-gram precision would be 3/5 and 1/5 (because “This” only occurs once in the reference, hence, it is the upper bound), respectively.

After having introduced the concepts of n-gram precision and *modified* n-gram precision, we can move to the definition of BLEU. Formally, BLEU is defined as the geometric mean of the *modified* n-gram precision of various length between a translation output and a set of references, multipled by a sentence brevity penalty which compensates the high precision we would expected from a short hypothesis with long references, i. e.:

$$BLEU = BP\left(\cdot\right) * \exp \sum_{n=1}^{N} \frac{\log p_n}{N} \tag{2.14}$$

where $BP(\cdot)$ is the brevity penalty, N is the highest n-gram order to consider and p_n is the *modified* n-gram precision.

Intuitively, the BLEU measures how many n-grams a translation output matches with a reference set.

2.10.2.3 NIST

NIST was introduced by [29] and it was proposed as an alternative to the BLEU metric. Instead of using the geometric mean of co-ocurrences, it uses the aritmetic average in order to reduce the big influence due to low co-occurrences for the larger values of N. Another difference is that while BLEU considers all n-gram equal during computation, NIST pre-computes how informative a n-gram is, regarding the references given, and weights it proportionally which in turn affects the final result.

2.10.2.4 METEOR

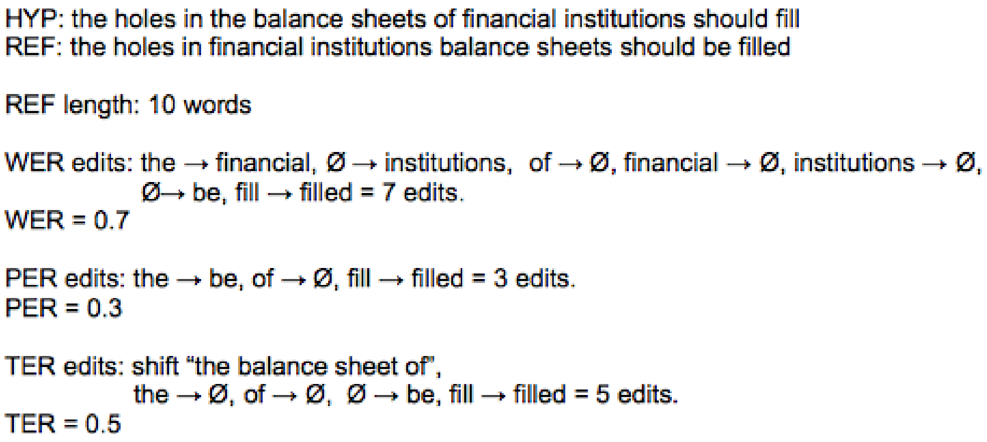
The METEOR metric [6] considers unigram precision and recall but also has some additional features like stemming and synonymy matching. We already defined unigram precision when talking about BLEU; unigram recall refers to the number of unigram found in the output which also occur in the reference, divided by the number of unigrams in the reference; and stemming is the process of reducing derived and inflected words to their stems (a stem is the part of a word that is share by its inflected forms), for instance, stemming the words "translate", "translation", "translated" would all result in "translat" (notice that the stem does not need to be a word by itself). These features make METEOR a computational expensive automatic metric which has helped BLEU become the standard in automatic metrics despite of METEOR's higher correlation with human evaluation.

2.10.2.5 Translation Edit Rate (TER)

This measure was proposed by the GALE (Global Autonomus Language Exploitation) project. It follows the idea behind WER: to determine the minimum number of edits between the hypothesis and the translation, but it consider a new type of edit: phrasal shifts. A phrasal shift moves a contiguos sequence of words from one position to another within the translation hypothesis. It follows the same equation as WER to compute its value but differs on the concept of edits. Figure 2.7 shows an example of WER, PER and TER.

$$TER = \frac{\# \, of \, edits}{avg. \, \# \, reference \, words} \tag{2.15}$$

where *edits* are deletion, replacements, insertions and shifts.



HYP: the holes in the balance sheets of financial institutions should fill
REF: the holes in financial institutions balance sheets should be filled

REF length: 10 words

WER edits: the → financial, Ø → institutions, of → Ø, financial → Ø, institutions → Ø,
Ø→ be, fill → filled = 7 edits.
WER = 0.7

PER edits: the → be, of → Ø, fill → filled = 3 edits.
PER = 0.3

TER edits: shift "the balance sheet of",
the → Ø, of → Ø, Ø → be, fill → filled = 5 edits.
TER = 0.5

Figure 2.7: Comparison between WER, PER and TER metrics

2.10.2.6 Human-targeted Translation Edit Rate (HTER)

TER compares the translation hypothesis with one or more pre-determined references but it ignores notions of semantic equivalence. It may be the case that these pre-determined reference are far from the hypothesis because of synonyms and therefore they would penalize it. The ideal situation would be that the comparison was made between the hypothesys and the reference that is closer to it, among all possible fluent references with the same meaning.

HTER is an evaluation metric based in TER that adds a man-in-the-loop component in the process. It was proposed by Snover [74] in 2006. With HTER, the human evaluator must generate a reference that satisfies the ideal situation described before. The new reference could be obtained either editing the pre-determine references or the translation hypothesis. The new reference must be fluent and it must have the same meaning of the pre-determined references. Once it is created, TER is computed between the translation hypothesis and the new reference.

In his research, Snover explains the process with the following example: suppose we have the translation hypothesys "The expert who requested anonymity said that the situation of the matter was linked to the dead bodies" and the pre-determined references "The expert, who asked not to be identified, added that this depended on the conditions of the bodies" and "The experts who asked to remain unnamed said that the matter was related to the state of the bodies.". It can be seen that the pre-determined references do not use the term "anonymity" but use other alternatives like "unnamed" and "not to be identified". The human evaluator, however generated the new reference "The expert who requested anonymity said that the matter was linked to the condition of the dead bodies" which preserves the meaning of the pre-determined references but used the terms "anonymity" and "linked" found in the hypothesis.

2.10.3 Evaluating word alignment: AER

In the case of word alignment, quality is measured in terms of precision, recall, F-measure and Align Error Rate (AER) [67]. Like the automatic metric for machine translation quality, AER requires a manually aligned gold standard as reference to be compared with the candidate alignment.

Similar to the definitions of n-gram precision found in the BLEU metric, we defined the precision of a candidate alignment given a reference as the number of links in the candidate alignment that occur in the reference divided by the total number of links in the candidate alignment. Also, similar to the definition of unigram recall found in the METEOR metric, we defined the recall of a candidate alignment given a reference as the number of links in the candidate alignment that occur in the reference divided by the total number of links in the reference. Finally the F-measure relates both concepts and is defined as the harmonic mean of precision and recall.

Because of the ambiguity in the manual alignment task, links are labeled as “Sure” or “Possible”. A link is considered “Sure” if the relation between the words it links to is unambiguous and it is considered only “Possible” is the relation is ambiguous. This labeling of links defines two subsets in the gold standard: the “Sure” set of links S and the “Possible” set P, which is defined as the union between set S and all links labeled as “Possible”. With these two sets identified and a candidate set of links A to evaluate, the precision of Possible and Sure (PR_P and PR_S), recall of Possible and Sure (R_P and R_S), F-measure of Possibile and Sure (F_P and F_S) andAER are defined as:

$$PR_P = \frac{|A \cap P|}{|A|} \tag{2.16}$$

$$PR_S = \frac{|A \cap S|}{|A|} \tag{2.17}$$

$$R_P = \frac{|A \cap P|}{|P|} \tag{2.18}$$

$$R_S = \frac{|A \cap S|}{|S|} \tag{2.19}$$

$$F_P = \frac{2PR_P R_P}{PR_P + R_P} \tag{2.20}$$

$$F_S = \frac{2PR_S R_S}{PR_S + R_S} \tag{2.21}$$

$$AER(S, P, A) = 1 - \frac{|A \cap S| + |A \cap P|}{|S| + |A|} \tag{2.22}$$

as it can be seen, according to equations (2.16) and (2.19) the AER metric relates the precision of the Possible set and the recall of the Sure set. As a final remark regarding AER, it is important to mention that even though word alignment plays an important role in SMT and AER is a common metric to measure its quality, there has not been any study which shows that decreases in alignment error rate (AER) always result in significant increases in translation performance [36].

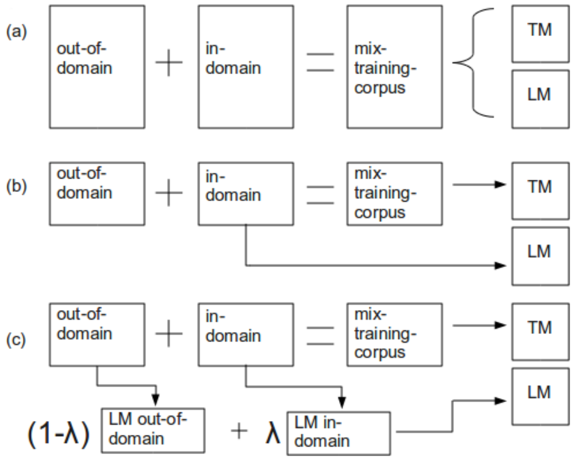


Figure 2.8: Concatenating corpora and model interpolation.

2.11 Domain adaptation

SMT systems are trained using parallel corpora. Therefore, once the system is trained and tuned, it is tightly coupled to the specific domain the training corpus belongs to. Text corpora can be different in vocabulary, style or grammar and a method to adapt to different domains is preferred than building a whole new system for each domain we may face. Moreover, it might be the only plausible solution as we may have a big out-of-domain parallel corpus but a small in-domain corpus which, if used alone, would perform poorly.

An example of a situation where these methods are required would be a SMT system trained with the Europarl Corpus [52] but used for translating movie reviews.

Different strategies can be used to perform such adaptation, and they all require a small in-domain corpus whether it is bilingual or not, for the system to adapt.

The following sections review some of these domain adaptation methods covering different approaches.

2.11.1 Combination of language and translation models.

A straightforward strategy would be to concatenate both corpora and train both the translation model and the language model from the resulting extended corpus. The idea in that case is to easily incorporate all the characteristics from the small in-domain corpus into the main out-of-domain corpus. Nevertheless, if the difference in size is relevant, the influence of the small corpus over the general system may not be as expected. Diagram (a) in Figure 2.8 shows this approach.

Another strategies may combine both corpora in different ways. For instance, besides training the translation model with the previous approach, the language model could use only in-domain corpora. The corresponding diagram of this strategy can be seen in Figure 2.8(b).

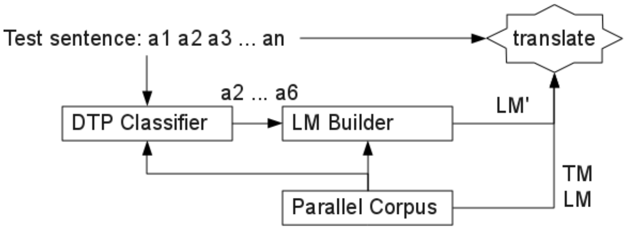


Figure 2.9: A SVM classifier identifies the DTPs and a LM is built accordingly. The rest of the sentence is translated with the global LM.

Finally, in case where in-domain corpus size may not be big enough to provide confidence probabilities and may have a small vocabulary, the language model could be obtained by a linear interpolation of the out-of-domain language model and a smaller in-domain language model. With this approach, the probability of a n-gram w would be:

$$P\left(w\right) = \lambda P_{in\_domain}\left(w\right) + \left(1-\lambda\right) P_{out\_of\_domain}\left(w\right) \tag{2.23}$$

where $P_{in\_domain}\left(w\right)$ represents the probability given by the language model trained with in-domain corpus and $P_{out\_of\_domain}\left(w\right)$ represents the probability given by the language model trained with out-of-domain corpus. The parameter λ would be set in order to minimize perplexity over a certain in-domain target language corpus. Diagram (c) in Figure 2.8 shows this approach. A comparison of these strategies is presented in [56].

The interpolation proposed by Koehn's research focused mainly on the language model. In general, language model adaptation has been well studied over the years ([21, 57, 2, 30]) both in speech recognition and machine translation.

Other technique that combines language models is based on the definition of Difficult-to-Translate phrases [62](DTP). This approach presented in [63], aimed at solving mainly two problems: disambiguating target language words and short distance word movements.

First, a classifier based on Support Vector Machines [27] is built to identify DTPs on the source language. Then, following the links given by the alignment we can identify the target language phrases that correspond to the source DTPs and select only the target language sentences which contain them to build a specific language model for that phrase. The rest of the sentence would be translated considering only the global language model. This strategy is shown in Figure 2.9.

Recently, there have been other studies that have focused on the translation model (TM) and how multiple TMs can be combine to achieve domain adaptation.

One of these research is Foster and Kuhn 2007 [35]. In their study, they applied a mixture modeling approach that consist on dividing the training material in different domains and training a model for each domain separately. After that first training is completed, the different models are weighted accordingly to the translation context. Benerjee et. al. [5] also follow that line in their study with User-Forum Data and using automatic classifiers [4]. This line of work is also followed by Sennrich [72] where he considers perplexity minimization to perform the combination of TMs similar to the idea behind LM combination.

The framework proposed by Foster and Kuhn is very similar to the one seen in Diagram (c) in Figure 2.8 but with the TM instead of LM. Later, Razmara et. al [70] proposed a model combination during decoding in a decoding framework that is similar to Chiang [17], which is based in a CKY parser.

Apart from weighting differently the respective TMs, others strategies consider adding more features to the log linear model of equation (2.5) [65]. All this cases, however, are based on different translation models for each domain. On the other hand, Wang et. al. [82] proposed a strategy that uses only one translation model with multiple weights and changes the decoding phase to be aware of the domain it is translating. In order to achieve this, during phrase extraction they recorded the domains where the phrases came from and used this information during decoding and tuning. During decoding, different λ values are used depending on the domain of the sentence, which is classified according to the phrases that are needed to translate it.

2.11.2 Using Information Retrieval.

Information retrieval (IR) is an area of computational linguistics that studies how to search information within documents. Techniques from IR have also been used to adapt a SMT when in-domain corpus is small. In 2005, Hildebrand [42] used a common measure in IR called *tf-idf* and the *cosine similarity* to select sentences from the training corpus that were closer to a given test set. Then used this adapted training corpus to build both the translation model and the language model.

The term *tf-idf* stands for "Term Frequency, Inverse Document Frequency". It is a statistical measure that weights how important a word is to a document in a specific corpus. It is commonly used to represent text documents in a vector space. Mathematically, for a given word i in document j, *tf-idf* is defined as:

$$tfidf\,(i,j) = tf\,(i,j) * idf\,(i) \tag{2.24}$$

where

$$\begin{aligned} tf\,(i,j) &= \frac{n_{i,j}}{\sum_k n_{k,j}} & (2.25) \\ idf\,(i) &= \log \frac{|D|}{|\{d : i \epsilon d\}|} & (2.26) \end{aligned}$$

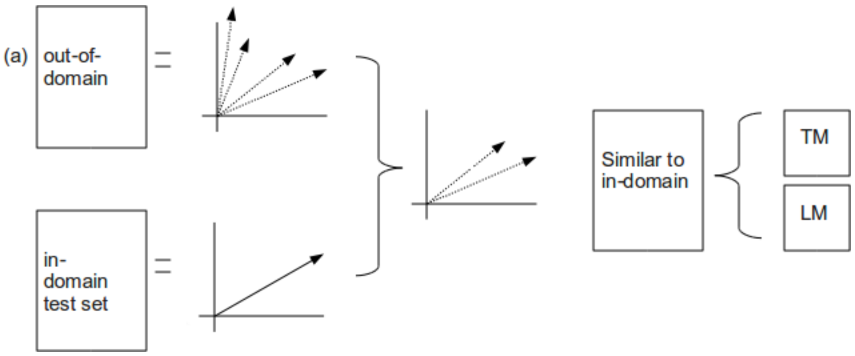


Figure 2.10: Using Information Retrieval for Domain Adaptation.

and $tf(i,j)$ represents the *term frequency* of word i in document j; $n_{i,j}$ counts the number of times word i appears in document j; $|D|$ is total number of documents; finally $|\{d : i\epsilon d\}|$ counts the number of documents where word i appears.

It can be seen from $tf\ (i,j)$ that words which appear often in a document are considered important for that document. On the other hand, very common words like articles and prepositions, tend to be frequent in a lot of documents, without being important. Hence, the term $idf(i)$ diminish their effect over the final weight.

Now, to represents documents in a vector space model, we define each document d as a vector $d = (w_1, w_2, \ldots, w_n)$ where each dimension correspond to a different word. If word w_i appears in document d, then its value in the vector will be $tfidf(i,j)$.

Once documents have been represented as vectors, the *cosine similarity* measure can be used to see how close is a document from another. The *cosine similarity* for vectors v and w is define as the cosine of the angle between the vectors:

$$cos\alpha = \frac{v * w}{\|v\| * \|w\|} \tag{2.27}$$

Its range is $[0,1]$ where zero means the vectors are orthogonal. Values closer to 1 mean more similarity between vectors.

In order to use the concepts from IR in SMT, each single sentence in the corpus is considered a separate document. Then, as proposed in [42], for each test sentence t from an in-domain corpus a subset of the n most similar sentences is extracted from the out-of-domain. Later, both translation and language model can be obtained from that extracted training corpus. Figure 2.10 models this procedure.

More recently, Banchs and Costa-jussà [3] also used cosine similarity for selecting the closest sentences in the training dataset to the input sentence to be translated, adapting the selection of translation units to the specific domain and context of the input sentence

Moving to another area of IR, Cross-lingual Information Retrieval (CLIR) was studied by Snover et. al. [73]. It was used to select documents in the target language from a monolingual in-domain corpus, using as query the source documents that were about to be translated.

The problem can be seen probabilistically as: given a query Q (a source sentence), find a document D (a target sentence) that maximize the equation $P(D|Q)$. This equation can be expanded using Bayes' Law as shown in (2.28). There, $P\left(Q\right)$ is constant for all documents and the prior probability $P\left(D\right)$ was considered uniform, hence constant for all D.

$$P(D|Q) = \frac{P\left(D\right) * P\left(Q|D\right)}{P\left(Q\right)} \tag{2.28}$$

Therefore, the problem is to find the document D that maximize $P\left(Q|D\right)$. A method for calculating such a probability was proposed in [86]. It considers every source language word f from query Q, the set of running words F in the source language corpus, and each target language word e.

$$P\left(Q|D\right) = \prod_{f \epsilon Q} \left[\alpha P\left(f|F\right) + \left(1-\alpha\right) \sum_{e} P\left(e|D\right) P\left(f|e\right)\right] \tag{2.29}$$

where α is constant, $P\left(f|F\right)$ if the probability to generate word f from set F, $P\left(e|D\right)$ is the probability to generate target language word e in document D and $P\left(f|e\right)$ is the probability of translating the word e into f.

Using equation (2.29) we can choose the N highest rank documents for each Q to adapt both the language model (with a linear interpolation similar to equation (2.23)) and the translation model, generating artificial phrases.

The process of generating artificial phrases starts by collecting for each query Q, only the phrases p_D that appear at least M times on the N highest rank documents for that query.

Then, for each phrase p_Q extracted from the source sentence (the query) a bilingual artificial phrase (p_Q, p_D) is created with a very low uniform probability as well as a lexical weight, which can be obtained from the out-of-domain bilingual corpus. Although this strategy will create a lot of wrong phrases, the language model will help to select the correct ones.

In 2007, another approach was proposed. For instance, to have a general-domain SMT system trained with all corpora but tuned with different in-domain development sets, hence having a different set of λ_i (see equation (2.5)) for each of them. The language model of this system was built using a mixture of K domain dependent language models. Therefore the probability of a given n-gram w was:

$$P\left(w\right) = \sum_{k=1}^{K} \gamma_k P_k\left(w\right) \tag{2.30}$$

where γ_k are the mixture weights and $P_k(w)$ is the language model trained with in-domain corpus k. It can be seen that this language model mixture is a generalization of equation (2.23). Just set $K = 2$, $(\gamma_1, \gamma_2) = (\alpha, 1 - \alpha)$ and P_1, P_2 to the in-domain and out-of-domain language models.

Now, in order to translate a test set, a text classification method is applied over it to determine its domain. Later, the specific set of λ_i is used to perform the translation. An information retrieval approach for text classification is proposed in [85], as well as perplexity tests over domain-dependent language models. The IR approach defines the similarity $S(d)$ between a test document and the development set of domain d:

$$S(d) = \sum_{w \epsilon A \cap A_d} \frac{1}{(|A^w| + 1) * |A|} \tag{2.31}$$

where A_d is the vocabulary of the development set of domain d, A is the vocabulary in the test document, $|A|$ its vocabulary size, and $|A^w|$ is the number of documents in the test set containing the word w. As the previous IR methods, the domain selected is the one with the highest $S(d)$ value.

2.11.3 Generating a Synthetic Corpus.

These methods require an in-domain source language corpus and optionally an in-domain translation dictionary besides the usual out-of-domain bilingual corpus. The idea is to train the SMT system with the out-of-domain corpus. If an in-domain translation dictionary is available, an initial set of probabilities can be assigned to each term and then it can be linearly interpolated (similar to equation (2.23)) with the translation model.

Once the first translation model is estimated, the in-domain corpus is translated in order to build a synthetic bilingual corpus that can be used to re-estimate the translation model. This process of translate and re-estimate is repeated until no improvement on a development set can be achieved. The estimation step can consider a full retraining of the original translation model, keep an additional translation model generated with the synthetic corpus as a new feature function (an additional model) or perform a linear interpolation between both translation models.

In [78] these different ways of estimation are considered and compared although the in-domain translation dictionary is not used at the beginning. Later on, in 2008, Wu extended that strategy with the in-domain translation dictionary and linear interpolation of the language model (in case a target language in-domain corpus is available). Figure 2.11 shows a diagram for the Synthetic corpus approach.

2.11.4 Automatic post-edition.

The previous methods were all designed for Phrase-based MT systems, but strategies for Rule-base MT systems have been proposed too. Isabelle et. al. [46] conducted experiments were a general domain

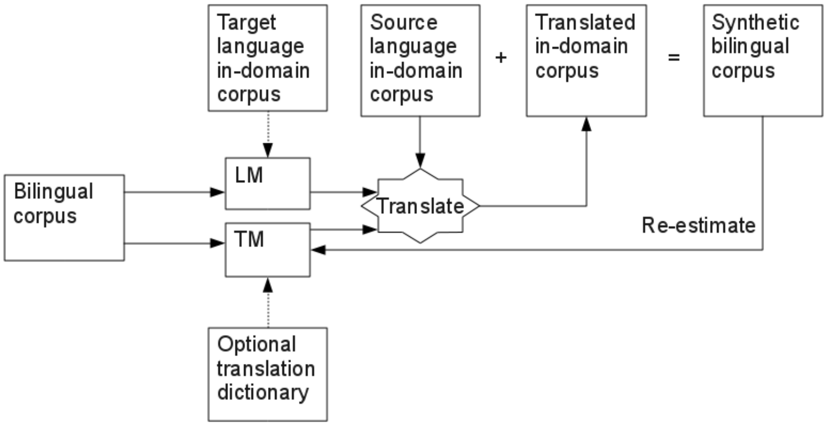

Target language in-domain corpus
Source language in-domain corpus
+
Translated in-domain corpus
=
Synthetic bilingual corpus
Bilingual corpus
LM
TM
Translate
Re-estimate
Optional translation dictionary

Figure 2.11: Generating synthetic corpus for domain adaptation

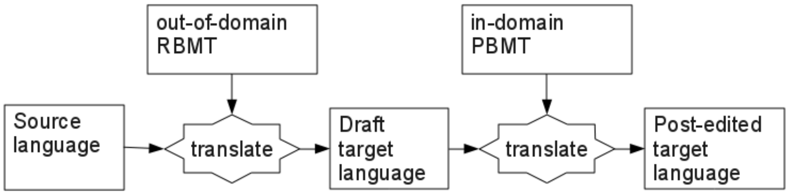


Figure 2.12: Automatic post-edition as a domain adaptation method.

Rule-base MT system is adapted to different domains using automatic post-edition with an in-domain trained Phrase-based MT system. The architecture consisted in a Rule-base MT system trained to translate general-domain sentences and a specific Phrase-based MT system trained to use as source language the previous output in order to correct it accordingly to the context it was trained for. The architecture of this system can be seen in Figure 2.12.

More recently, Rubino et. al. [71] also studied automatic post-edition with two differences from the precious work: the framework consisted of two Phrase-based SMT instead of one Rule-based and one Phrase-based, and they added a sentence classifier component that determines which sentences will form the in-domain data for the post-edition step.

2.11.5 Active Learning

Active Learning is a machine learning technique in which the algorithm iterates over an unlabeled set of data, choosing a subset from it and asking for labels in order to continue learning. It is specially useful in areas where labelled data is hard to acquire but unlabelled is not. For Machine Translation, we can consider bilingual corpora to be labelled data and source language corpora to be unlabeled.

An strategy for domain adaptation based on the concept of Active Learning can be found in [41]. The proposal can be summarized as follows: a baseline SMT system is trained and used to generate a synthetic parallel corpus from a source language corpus and an additional translation model is built from the synthetic corpus; then, k sentence pairs from the synthetic corpus are removed and their translations are replaced with human translations. Once the replacement is done, the k sentence pairs are added again to the synthetic corpus and the models are reestimated. This process continues until no improvement is observed over a test set.

2.12 User feedback

Traditionally, user feedback is associated with commercial web applications. For instance, you can write book reviews on Amazon and rate items so the system can propose you new products according to your profile. As a music shop, Spotify looks over your music history to determine the music you like and show you similar artists; you can also rate your songs to bias the recommendations. These characteristics are possible thanks to recommender systems. A review of these strategies is presented in [1]. Based on how recommendations are made, they can be categorized as follow:

Content-based recommendations: Where the user will be recommended items similar to the ones the user preferred in the past.

Collaborative recommendations: Where the user will be recommended items that people with similar tastes and preferences liked in the past.

Hybrid approaches: Where a combination of the previous two is presented.

Coming back to machine translation, we can draw an analogy between these approaches in commercial web applications and using user feedback to improve a MT system. On one hand, the users will receive better translations learnt from suggestions made by other users, like collaborative recommendations give results related with users with similar tastes; on the other hand, the system can adapt itself to the writing style of a specific user using her feedback (for instance, it can learn to use the word “lift” instead of “elevator” if the user is from England), just like content-based recommendation show results depending on your past preferences.

User feedback in machine translation is found in most web translation systems. For instance, Google Translate[2] allows the user to suggest an better translation than the one provided and Microsoft’s Bing Translator[3] allows the user to rank the provided translation. Moreover, Google has a patent about their feedback system [20], it mentions possible filters that can be applied to user’s input and the ability to automatically provide the alternate translation to a human reviewer.

As research contributions, recently Dobrinkat considered user feedback to build a bilingual corpus for domain adaptation [28]. It developed a system which loaded a monolingual corpus and asked their users to provide translations and to evaluate them. After that, the generated bilingual corpus was used to perform domain adaptation over an out-of-domain Phrase-based MT system.

[2] http://translate.google.com

[3] http://www.microsofttranslator.com

Between 2010 and 2013, the FAUST project contributed extensively with researches in the area of User Feedback and Machine Translation. In this context, the User Feedback problem can be divided in two different phases: first, we need to collect user feedback and filter out all post-editions that can be considered noise, for instance post-editions with grammatical errors, post-editions that are identical to their translation or post-editions that are worst than the output obtained by the system; second, once we have a collection of usefull feedback we need to defined a strategy to incorporate the collected information into the baseline system, focusing on improving its translation quality.

Researches on the first phase include [69], where the authors analyze a corpus of user feedback to study the different post-editions users provide. More recently, Barrón Cedeño et. al. [7] focused on the task of filtering user feedback collected from a real weblog with the purpose of improving a SMT system. In their study, which followed a previous one made by Pighin et. al. [69], the research team trained a Support Vector Machine to perform a binary classification to identify the feedback that was better than the automatic translation, in terms of some human evaluation guidelines proposed in the research. The feature vector was built using the information from the input sentences, the automatic translation, the user feedback, and the back translation of automatic output and feedback to the source language. A subset of these features is: length in tokens of each sentence, Levenshtein distance between translation and feedback divided by the length of the former, number of words in translation not in feedback divided by number of words in translation, a binary feature to indicate if the input sentence contained up to five words and character 3-gram cosine similarity between all pairwaise choices from input, output, feedback and back translations. The training data came from weblogs of the Reverso SMT [4] system and it was manually annotated to separate useful feedback (according to some annotation guidelines defined) and noise. The level of agreement between the annotators in terms of Cohen's kappa coefficient [22] was, as the authors claim "moderately high but not fully satisfactory", indicating the particular difficulty of the task, even for human experts.

The FAUST project also contributes with researches focused on general strategies for improving SMT systems. For instance, in [33] a character-based translator is used as preprocess to fix misspeled words in the input and [34] proposes a strategy based on generalization and generation of morphology.

As part of the FAUST project, this thesis focuses on the second phase of User Feedback: given a collection of usefull post-editions provided by users, we have defined a strategy to incorporate the collected information into the baseline system, focusing on improving its translation quality. The following two chapters describe in detail the proposed strategy and the different scenarios where the it has been tested.

[4] http://www.reverso.net

Chapter 3

Translation-based Word Alignment

In this chapter we present the concept of translation-based word alignment. We begin defining the notion of Augmented Corpus, a parallel corpus expanded with translated sentences obtained with a baseline MT system. Then, we explore the idea of Change Zones and how they can help us to identify useful translation units and critical points towards a chunk-based alignment. After that, we generalize the concept to obtain word alignments that can be used in different SMT paradigms.

The main objective of the ideas presented here is to improve a baseline system, focusing on fixing the translation errors it makes during decoding. We will see the obstacles that appear in each phase of the process and how they can be solved in order to arrive to a final solution.

By the end of the chapter, we will have all the necessary information to apply the same strategy to different scenarios.

3.1 Augmented corpus

Suppose we have a previously trained SMT system that has an acceptable performance in the domain it was designed for. We would like to adapt that system to scenarios it has not seen, for instance it was trained to translate Parliament sessions and we would like to translate news articles or tourist dialogues; another examples could be that we would like to renew its vocabulary coverage and writing style or that we would like to correct errors we have seen during translation by using user's feedback.

In order to adapt or to correct our system we have available a small bilingual corpus (a few thousands sentences) specific to the problem we plan to solve. Besides the additional bilingual corpus we also have the translation output of its source side computed with the system we want to adapt. Therefore our new data, named *Augmented Corpus* has actually three parts: The source side of the bilingual corpus (called input), the translation output (computed with the trained system) and the reference (the target side of the bilingual corpus). A graphical description of this material can be seen in Figure 3.1.

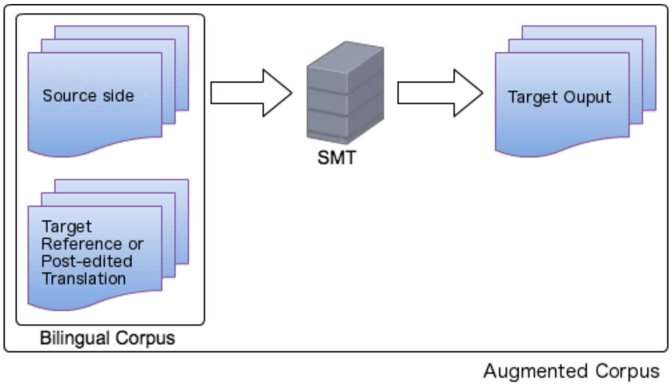


Figure 3.1: Augmented corpus composition

An Augmented Corpus can be obtained in two different ways, each of them equally valid to gather material to improve an SMT system.

The first option would be a domain adaptation scenario. Suppose we have a out-of-domain baseline system and we also have available a small in-domain parallel corpus usefull for adaptation. We could augment the in-domain corpus including the baseline translation of its source side, completing the three parts of an Augmented Corpus.

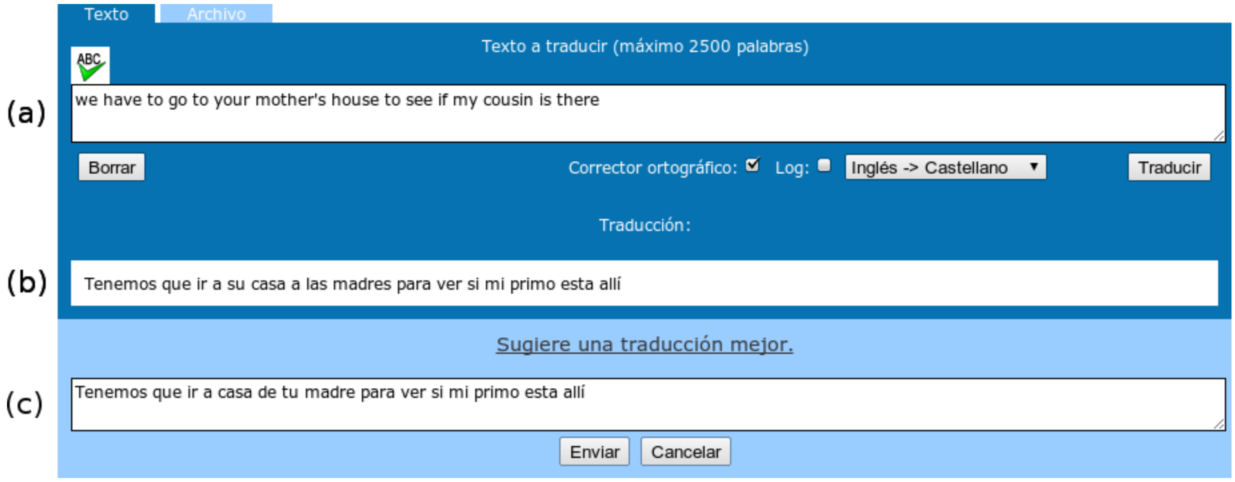
The second option to build an Augmented Corpus is with user's feedback. In this case, instead of originally having the input and the reference, we obtain first the input and its translation output; then, the user can complete the corpus with her feedback, i.e. with a postedited version of the translation computed by the decoder.

3.1.1 Consideration About Using User's Feedback

Building an Augmented Corpus with user's feedback has its inconvenience, specially regarding the feedback's quality. In spite of the frequent use of online machine translators, users do not tend to send feedback for system improvement, and even when they do, it is hardly useful. Usually, the system offers the functionality of sending feedback without restrain. Therefore, feedback collecting algorithms must confront with vicious feedbacks, orthographic errors, text unrelated with the original query, etc.

For that reason, to exploit user's feedback we have to deal with two different problems: how to filter user's feedback so we keep only the valuable data; and how to use the selected data to improve the machine translation system. The FAUST project was created with the objective of solving these problems. Among the project contributions that address the first problem, we find the work of Pighin et. al., analysing the different responses to MT output [69] and the filtering strategy proposed by Barrón Cedeño et. al. [7].

In this thesis, we are focused on the latter task and therefore we assume we can build an Augmented Corpus with valuable information in order to be used in the following sections.


Texto
Archivo
Texto a traducir (máximo 2500 palabras)
ABC
(a)
Borrar
Traducir
Traducción:
(b)
Sugiere una traducción mejor.
(c)
Enviar
Cancelar

we have to go to your mother's house to see if my cousin is there

Corrector ortográfico: ☑ Log: □ Inglés -> Castellano ▼

Tenemos que ir a su casa a las madres para ver si mi primo esta allí

Tenemos que ir a casa de tu madre para ver si mi primo esta allí

Figure 3.2: Post-edition Example showing (a) the input sentence (b) translation output and (c) post-edited output to be used as reference

3.2 Change Zones

Suppose we have collected an augmented corpus from user's feedback. Therefore, we have the following information available: input sentences, their translation outputs and their valid post-edited versions (references) which aim to correct possible errors that the system made during decoding.

The idea now is to compare these references with their corresponding translation outputs, in order to determine where the decoder made a mistake. This comparison could allow us obtain an alignment between input and reference that could be used to extract new translation units that would benefit the baseline system.

To design a strategy with that idea in mind, Figure 3.2 shows a real example, extracted from the English to Spanish SMT system N-II[1]. The image also shows the English input sentence that the system was translating.

Intuitively, it can be seen that the difference between the output and the reference "is small". The decoder translated most of the sentence correctly, but it made a mistake with the phrase "your mother's house" choosing "su casa a las madres" instead of "casa de tu madre". The first step then, is finding an algorithm that detects automatically these spans so it can correct them without altering the rest of the translation.

What we need is a string similarity metric to translate that "is small" intuiton into a formal number. For that, we introduce the standard edit distance or Levenshtein distance. The Levensthein distance is defined as the minimum number of edits (insertions, deletions and replacements) needed to transform one string into another. It is originally a character-based measure but it can be easily used for sentences as well, considering the words as characters.

[1] http://www.n-ii.org

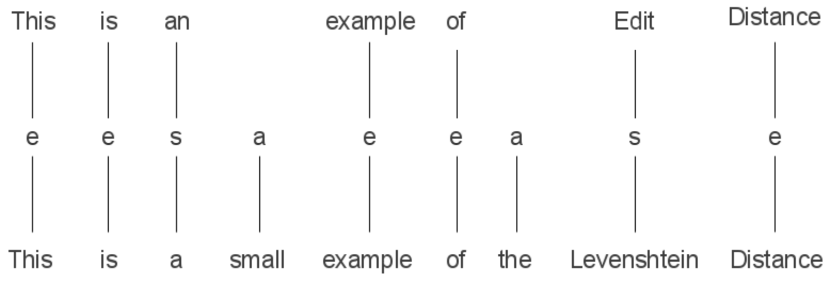


Figure 3.3: Levenshtein Distance example with its corresponding Edit Path

When computing the Levenshtein distance between two strings, together with the minimum number of edits we also obtain an edit path, i.e., the sequence of changes needed. Figure 3.3 shows an example of a Levenshtein distance between the sentences "This is an example of Edit Distance" and "This is a small example of the Levenshtein Distance" and its edit path in between. In that case, the Levenshtein distance between the two sentences is 4, which corresponds to one replacement (represented by letter s), two insertions (represented by a) and one final replacement. Deletions, altough not in this example, are represented by the letter d while letter e represents words that do not change between sentences ("e" for "equal").

After looking at the edit path in Figure 3.3 we can see that changes are represented by consecutive letters that involve something other than an "e". We call this consecutive letters, or spans, "change zones". Formally, we defined the change zones of an edit path p as the longest substrings of p that matches any of the following regular expressions:

$$[sda] * s\,[sda] * \tag{3.1}$$

$$e\,[da] + \tag{3.2}$$

$$\hat{}[da]\,e \tag{3.3}$$

continuing with the example shown in Figure 3.3, the change zones for that pair are sa and as, and the words involved in those changes form two different correction phrases, (an → a small) and (Edit → the Levenshtein). It is important to mention that the expressions (3.2) and (3.3) include an "e" in their pattern in order to always have words from the output and the reference in the corresponding correction phrases. Finally, if a substring p matches more than one expression, we will consider the change zone represented by the first of the matches, according to the order provided above.

By using change zones we are now able to automatically detect where the post-edited corrections happened. However, our main objective, as defined at the beginning of this section, is to produce a input-reference alignment. Therefore, we still need to find this correspondence.

The idea of identifying these spans is to combine all words from a change zone in a single chunk and keep the remaining decoded sentence intact. In this way, we will preserve the translation units that

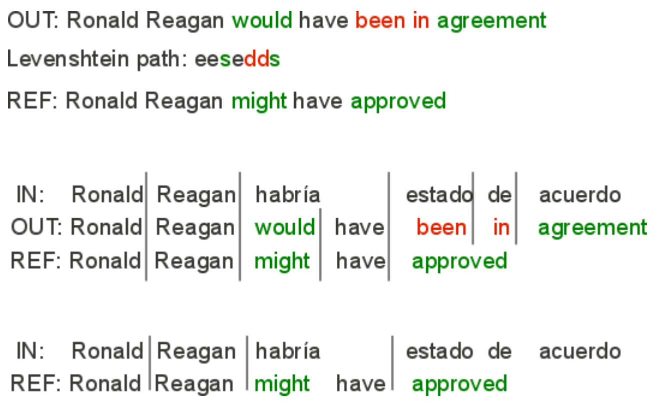

OUT: Ronald Reagan would have been in agreement
Levenshtein path: eesedds
REF: Ronald Reagan might have approved
IN: Ronald | Reagan | habría | estado | de | acuerdo
OUT: Ronald | Reagan | would | have | been | in | agreement
REF: Ronald | Reagan | might | have | approved
IN: Ronald | Reagan | habría | estado de acuerdo
REF: Ronald | Reagan | might have | approved

Figure 3.4: Example of the extraction process. (i) First we compute the Levenshtein path between output and post-edition (change zones and related words are colored), (ii) then we segment the pair input-reference considering the original units and the changed zones. (iii) Shows the final segmentation with the new units added.

were used correctly and we will replace the ones that produced the difference between the translation output and its reference. The result will be a chunk-based alignment of the parallel sentence.

The process would be as follows: first, we compute the translation to obtain the units used during decoding. Then, we compare the translation output with the reference and detect the change zones, if any. Finally, we extract the same units used during decoding and replace translation output words by reference words according to the changed zones found. If a changed zone affects more than one translation unit at the same time, we join the affected units into a single unit and then proceed with the word replacement.

For instance, suppose that we have now a Spanish-to-English SMT system that translated the sentence "Ronald Reagan habría estado de acuerdo". The system produced "Ronald Reagan would have been in agreement" and as a reference we have "Ronald Reagan might have approved". A graphic example of the whole process can be seen in Figure 3.4. With this sentence pair and its translation output, we computed the Levenshtein distance, which is 4, and the Levenshtein path $p = eesedds$. The path indicates that the third word must be replaced, the fifth and sixth deleted and the last one replaced. According to the regular expression (3.1), this example gives us two different change zones, $s1 = s$ at the middle and $s2 = dds$ at the end; from $s1$ we have the correction phrase ($would \rightarrow might$) and from $s2$ we have ($been\ in\ agreement \rightarrow approved$). Finally, following the original units used during decoding, we change a word from the third unit and join the last three units into a single one, because their output words are involved in $s2$. All other units remain intact. The final chunk-baesd alignment can be seen at the bottom of the figure.

3.3 From Change Zones To Word Alignments

Because of the nature of the Levenshtein distance, if the reference differs much from the translation output, the chunks produced by the change zones will be innacurate. We can see an example at

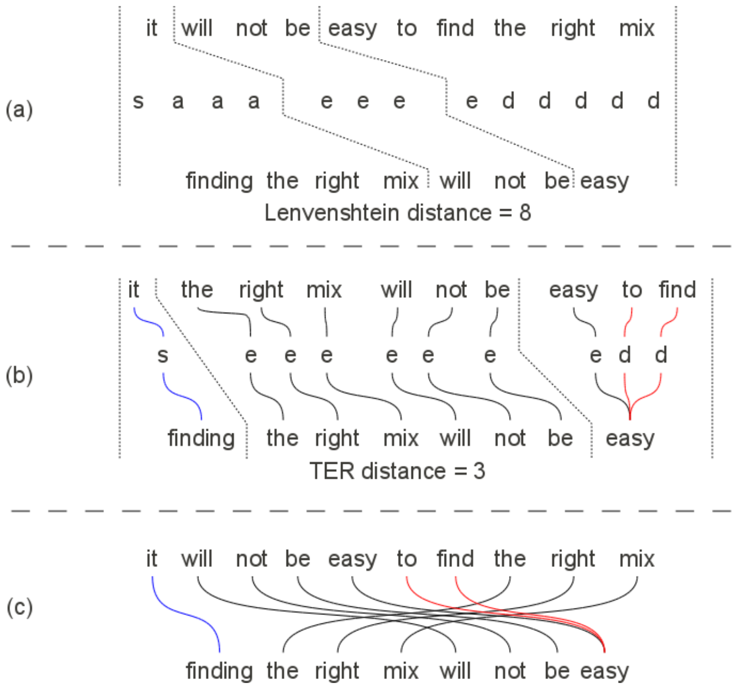

(a)
it will not be easy to find the right mix
s a a a e e e e d d d d d
finding the right mix will not be easy
Lenvenshtein distance = 8
(b)
it the right mix will not be easy to find
s e e e e e e e d d
finding the right mix will not be easy
TER distance = 3
(c)
it will not be easy to find the right mix
finding the right mix will not be easy

Figure 3.5: Comparison between Levenshtein and TER distances. (a) and (b) are the result of both methods, respectively. We can see how the movements in (b) improved the alignment. (c) illustrates the same links in (b) but with the words in their original places.

the top of Figure 3.5. There, the change zones *saaa* and *eddddd* produce the correction phrases (*it* → *finding the right mix*) and (*easy to find the right mix* → *easy*) which do not have valuable information to correct the baseline system.

Therefore we need a new theoretical metric that would be able to deal with word movements. First defined in [74], Translation Edit Rate (TER) is an error metric that measures the number of edits required to change a translation output into one of its references. It is based on the Levenshtein distance but it also allows word movement to obtain lower distances. The edit paths resulting from TER are able to align phrases that are far appart between the two sentences. An example of both edit paths in the same sentence can be seen in sections (a) and (b) of Figure 3.5, where the TER distance is shorter because of its ability to arrange words in order to reduce as much as possible the result. Section (c) in the image shows the alignment after the words were moved back to their original positions.

There is an external research, posterior to the Derived Units strategy and experiments described in this thesis, which is very similar in concept to the ideas described here. In that work, Blain et. al. [11] also proposed a comparison between translation and post-editions in order to build a word alignment between input and post-edition. Their proposal also used TER as a comparison tool to obtain the alignment between translation and post-edition.

Coming back to our study, even though changing from the standard Levenshtein distance to the TER distance intuitively improves the detection of changed zones, we will still misalign many words if we rely only on it. In the example from Figure 3.5 the verb “to find” from the output sentence will be incorrectly grouped with the previous word “easy”, according to the regular expression (3.2), resulting in the correction phrase ($easy\ to\ find \rightarrow easy$). Also, the pronoum “it” will be aligned with “finding” which is undesirable. The ideal case would be aligning ($to\ find \rightarrow finding$) and moving “it” together with “will not” to produce ($it\ will\ not \rightarrow will\ not$).

The problem comes from the fact that the words related with an edit (either adding, deleting and replacing) are still aligned according to their positions in the sentence and not by their semantic meaning or context.

In order to find a solution for this problem, we reformulate the word alignment method between an output translation and its reference as a two-step process: first, we compute the TER edit path between the translation output and its reference in order to detect the words that are identical in both sentences (i.e., all e steps in the path) and we link them together; then, we deal with the remaining words, considering all possible combinations between output and reference words and aligning the best possible pairs. This two-step process implementation allows for using the results from the first step as a constraint for selecting the links of the remaining words.

3.4 Word Alignment In An Augmented Corpus

Computing a word alignment in an Augmented Corpus is another way of extracting new translation units. Using word alignments is also preferable to edit distances as the former do not force words to stay together the way change zones do. We can see that again in Figure 3.5. In the figure, we have the same sentence with different information: section (a) shows the change zones detected; these zones forced the output words “easy to find the right mix” to stay together in the same unit, even though its corresponding word “easy” in the reference does not provide all the information; a word alignment like the one in section (c) of the same figure is preferable. Even though we still have some errors from the TER computation and aligmnent in (b) (errors we aim to solve in the following subsections), we do not force the words to stay together in this step if we do not have a direct link between them.

In the previous section, we introduced the idea of a two-step process implementation. The first one was already defined: to compute the TER between the translation output and its reference, and align all words that are identical in both (i.e. words corresponding to an “e” step in the edit path). In Figure 3.5, this would mean removing the wrong blue and red links. The second step is described in the following subsection, where a similarity function is defined and used to link the remaining words between the translation output and the reference. Finally, two subsections explain the tuning process required for the proposed similarity function to work, and compare this method in terms of alignment quality against other alternatives.

3.4.1 A Similarity Function To Align Translations To References

In order to align words between the translation output and its reference, we need a function $Sim(w_{t,}w_r)$ that measures the similarity between an output word w_t and a reference word w_r. With the function defined, we iterate from left to right through all non-aligned output words; for each word w_t and word w_r we compute $Sim(w_{t,}w_r)$ and we assign a link to the pair with the maximum value:

$$link(w_t, w_r) \equiv w_r = \arg\max_{w'_r} Sim(w_t, w'_r) \tag{3.4}$$

After computing the alignment between the target sentence and its reference, the final alignment between source sentence and reference will be defined in the following way: there will be a link between a source word w_s and a reference word w_r if and only if there is a target output word w_t such that there is a link between w_s and w_t and a link between w_t and w_r.

$$link(w_s, w_r) \equiv (\exists w_t,\ link(w_s, w_t) \wedge link(w_t, w_r)) \tag{3.5}$$

Now we will focus on the similarity funtion needed to align the target sentence with its reference. We define $Sim(w_{t,}w_r)$ as a linear combination of eight different features:

$$Sim(w_t, w_r) = \left[\sum_{i=1}^{8} \lambda_i h_i(w_t, w_r)\right] \tag{3.6}$$

where w_t and w_r are an output word and a reference word respectively, $h_i(w_t, w_r)$ are the features involved and λ_i are the contribution weights of those features.

The following sections describe the different features h_i in $Sim(w_t, w_r)$, the special treatment of unknown words and the way to compute the contribution weights λ_i.

3.4.1.1 A binary feature for identical words

It might be the case that some words that are identical between target and reference were not aligned during the first step. This feature, h_1 was designed to consider them

$$h_1(w_t, w_r) = \begin{cases} 1 & , \ w_t = w_r \\ 0 & , \ otherwise \end{cases} \tag{3.7}$$

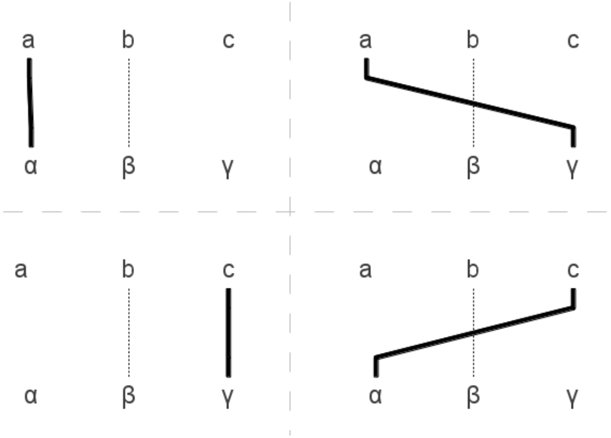


Figure 3.6: h_2 and h_3 are designed to detect neighbor links. A neighbouring link (bold line) may be an indicator of the current link (dotted line)

3.4.1.2 Two binary features to consider the context

Intuitively, if a target word w_{t-1} is already aligned with word w_{r-1} or w_{r+1} is likely that the next word w_t is aligned with reference word w_r. Similarly, we could say the same if we observe a link between w_{t+1} and one of the neighbors of w_r. The idea behind these features can be seen in Figure 3.6 and they are formally defined as follow:

$$h_2(w_t, w_r) = \begin{cases} 1 & , link(w_{t-1}, w_{r-1}) \vee \\ & \quad link(w_{t-1}, w_{r+1}) \\ 0 & \quad , otherwise \end{cases} \tag{3.8}$$

$$h_3(w_t, w_r) = \begin{cases} 1 & , link(w_{t+1}, w_{r-1}) \vee \\ & \quad link(w_{t+1}, w_{r+1}) \\ 0 & \quad , otherwise \end{cases} \tag{3.9}$$

3.4.1.3 A penalty feature for distant links

We do not expect a reference word w_r being aligned with two target words w_t and w'_t that are distant from each other in the translation output, hence we have designed a feature that penalizes these cases proportionally to the distance between w_t and w'_t within the sentence. Figure 3.7 shows an example of the idea. The formal definition of this feature is:

$$h_4(w_t, w_r) = \min_{link(w'_t, w_r)} \left(- \left\| \frac{pos(w_t) - pos(w'_t)}{N} \right\| \right) \tag{3.10}$$

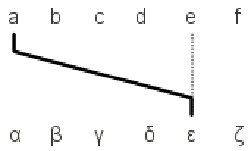


Figure 3.7: If a reference word $w_r = \epsilon$ is already aligned with a target word $w'_t = a$, feature h_4 penalizes the similarity between $w_t = e$ and w_r according to the distance between w_t and w'_t.

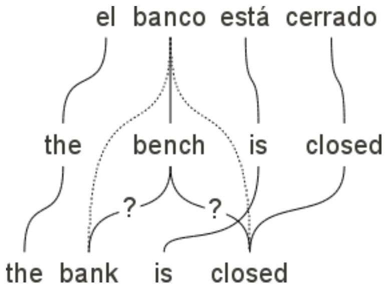


Figure 3.8: For $w_t = bench$, $w_r = bank$, and $w'_r = closed$, $h_5(w_t, w_r)$ is higher than $h_5(w_t, w'_r)$ because the dictionary gives a higher value to $LEX(banco, bank)$ than to $LEX(banco, closed)$.

where $pos(w_t)$ is the position of word w_t in the translation output and N is the number of words in the translation output. N is used for normalization purposes..

3.4.1.4 Lexical features that consider source words

The previous four features have only considered the position of words and the neighbouring links. Feature h_1 considers if the target and reference words are identical but it does not consider if they are somehow related (they might be synonyms or they might represent two different meanings of the same source word). The next two features were designed to take these cases into account. These features use the source sentence words along with a bilingual dictionary extracted from the baseline train data.

The idea behind these features is the following: let h_5 and h_6 be the features we are defining in this subsection and let $LEX(w_s, w_r)$ and $LEX(w_r, w_s)$ be discrete functions $(word, word) \to \mathbb{R}$ representing the bilingual dictionary learnt from the baseline train data; if a target word w_t comes from a source word w_s, features $h_5(w_t, w_r)$ and $h_6(w_t, w_r)$ will measure the degree of correspondence between w_t and w_r, based on the correspondences between w_s and w_r as defined by the bilingual dictionary. For instance, the Spanish word "banco" can be translated into "bank" or "bench" depending on the context but never into "closed", therefore $h_5(bank, bench)$ should be higher that $h_5(bank, closed)$ just like we expect higher values for $LEX(banco, bank)$ and $LEX(banco, bench)$ than for $LEX(banco, closed)$ in the bilingual dictionary. An example of these features can be seen in Figure 3.8

Besides detecting semantic relationships between words with different meanings (because of their common source word), these features will also detect the relationship between different forms of the same verb because of similar reasons.

Finally, because the bilingual dictionary is not symmetric with respect to the direction (source-to-target or target-to-source) considered for its computation, consistently, we have also defined two different features, one for each dictionary direction:

$$h_5(w_t, w_r) = \frac{\sum_{w_s:link(w_s,w_t)} LEX(w_s, w_r)}{\sum_{w'_r} \sum_{w_s:link(w_s,w_t)} LEX(w_s, w'_r)} \tag{3.11}$$

$$h_6(w_t, w_r) = \frac{\sum_{w_s:link(w_s,w_t)} LEX(w_r, w_s)}{\sum_{w'_r} \sum_{w_s:link(w_s,w_t)} LEX(w_r, w'_s)} \tag{3.12}$$

where $LEX(w_s, w_r)$ and $LEX(w_r, w_s)$ are the bilingual dictionaries obtained from the baseline training data. In our implementation, we use the maximum likelihood lexical weights provided by the Moses toolkit [55] as the bilingual dictionaries.

3.4.1.5 Special Features For Unknown Words

If one of the input words is unknown for the baseline system, it is not translated but simply duplicated in the translation output. The most common case for unknown words are proper and common names. An unknown word could also come from an inflected form that was not seen in the training corpus.

The similarity funtion defined so far has a problem with unknown words. The lexical features always return a zero value, because the input word is not in the bilingual dictionary and, unless it corresponds to a name and it appears in the reference as well, h_1 will also return a zero value for it. Therefore, we define two conditional features h_7 and h_8 that apply only for unknown words. They are defined as follows:

$$h_7(w_t, w_r) = \begin{cases} 1 - \frac{distance(w_t,w_r)}{|w_r|} & , if \quad w_t \neq w_r \\ \infty & , if \quad w_t = w_r \end{cases} \tag{3.13}$$

where $|w_r|$ is the length of word w_r and $distance(w_t, w_r)$ is the character based Levenshtein Distance between the two words. It is important to know that $h_7(w_t, w_r)$ could be negative in the first case, but it is not relevant to our purpose. Also note that if the words are the same, we force them to be linked with an infinite value.

Finally, $h_8(w_t, w_r)$ automatically discards the posibility of a link between w_t and w_r if w_t is an unknown word and w_r belongs to a list of stop words. To force this decision we simply assign minus infinity when the case occur. Formally:

$$h_8(w_t, w_r) = -\infty, \quad if \quad w_r \in SW \tag{3.14}$$

where SW is a set of stop words consisting of determiners, articles, pronouns, prepositions and very common verbs like, in English, “to be” or “to have”.

3.4.2 Computing The Contribution Weights

So far, we have defined eight different features that will help us determine the similarity between two known words. These features were designed to consider different cases where the words tend to be linked. However, these features correspond to different types of variables: some are binary features, one is a negative real value in the range $(-1, 0)$ and others are real numbers. In order to combine them properly and assure the best alignment quality we must learn the contribution weights λ_i of Equation (3.6).

This process can be seen as an optimization problem. Given a golden word alignment G between two set of sentences S_s and S_r, we would like to set the weights λ_i such that after obtaining a word alignment A, as described before in Equation (3.5), the Align Error Rate $AER(G, A)$ is minimum. Details about AER and how it is obtained can be seen in Section 2.10.3.

Once we have learnt the contribution weights λ_i, we are ready to obtain the word alignment from an Augmented Corpus in order to extract new translation units that will enhanced our SMT system.

3.4.2.1 Computing The Contribution Weights For A English-to-Spanish SMT System

In this scenario, we used two parallel corpora, both from the same domain. To train the baseline system we used the Europarl’s parallel corpus described in Appendix A.2. Appart from it, we need a manually aligned parallel corpus for tuning and testing the λ_i weights of our feature functions, as described before. We have chosen to work with another subset of the Europarl, not included in the baseline corpus, that consist of 100 sentences for tuning and 400 sentences for testing. You will find a full description of this material in Appendix A.2 as well.

The experiment proceeded as follows:

1. Train and tune the baseline system using the parallel corpora described above.
2. Translate the source side of the manually aligned corpus with the baseline system trained in the firts step.
3. Using the translation output and the manually aligned corpus together as an Augmented Corpus, a downhill Simplex algorithm is applied to set the weights that minimizes the AER.
4. Once the optimization process has finished, the aligning method is tested with the test corpus in order to determine its final AER.

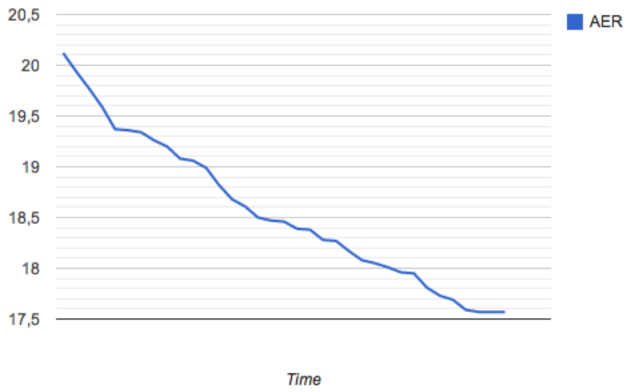


Figure 3.9: AER variation during downhill Simplex

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
0.08	0.75	0.91	3.08	0.47	2.02	1.5	1

Table 3.1: Contribution weights for the similarity function's features

The result of the optimization process (step 3) can be seen in Figure 3.9. Starting with uniform weights $\lambda_i = 1$ we were able to reduce the AER in 13% or 2.5 points from 20.12 to 17.57. The final weights can be seen in Table 3.1. The most important feature, according to their weights, are the penalization for distant links and one of the lexical features. It is curious that the "same word" feature for known words did not prove to be relevant compared with the others but this can be explained by the fact that it is somehow overlapped with the lexical feature, i.e. if the words are identical their lexical similarity should be high as well.

3.4.3 Measuring The Word Alignment Performance

Even though our final objective is to improve the translation quality of a SMT system, we can also evaluate the alignment process on its own, comparing it with similar approaches in terms of AER. For that comparison we have chosen the state-of-the-art alignment toolkit MGIZA [38].

We have already mentioned that, after optimizing the λ_i values, the AER dropped on the development set 13% from its original value with all $\lambda_i = 1$. Now we are going to present the result for different heuristics of word alignments over the 400 sentences of test set using the λ_i values of Table 3.1.

It is important to notice that our word alignment method relies on the links between input and output provided by the baseline system. The similarity function uses that preliminary result to compute an alignment between output and reference and, then, those two alignments are combined to produce the final input-to-reference alignment.

Therefore, we can obtain different input-to-reference results if we vary the alignment between output and reference.

	out-ref	ref-out	union	intersection	grow diagonal
AER	20.52	21.64	23.41	18.41	21.08
SURE precision	75.73	72.45	66.11	85.26	70.74
SURE recall	76.13	76.92	79.62	73.26	78.83
SURE F-measure	75.93	74.62	72.24	78.81	74.57
POSSIBLE precision	82.81	79.72	74.08	91.29	78.99
POSSIBLE recall	58.01	58.97	62.16	54.65	61.34
POSSIBLE F-measure	68.22	67.79	67.6	68.37	69.05

Table 3.2: Alignment performance using the Similarity Function

	in-ref	ref-in	union	intersection	grow diagonal
AER	18.79	19.88	19.77	18.82	19.06
SURE precision	79.9	76.96	69.58	91.77	71.95
SURE recall	74.9	76.02	81.42	69.5	80.49
SURE F-measure	77.22	76.49	75.04	79.09	75.99
POSSIBLE precision	87.93	84.26	79.21	96.62	81.35
POSSIBLE recall	57.02	57.44	63.97	50.5	62.8
POSSIBLE F-measure	69.18	68.31	70.78	66.33	70.88

Table 3.3: Alignment performance using MGIZA

With traditional alignment tools like GIZA++, the final result is usually a symmetrization of two alignments: input-to-reference and reference-to-input. Different symmetrizations are union, intersection and grow diagonal [54].

In our case, without changing the input-to-output alignment provided by the system, the similarity function iterates over the output sentence and for each output word, it assigns a link to its most similar reference word. This method assures that all output words are aligned to some reference word but some of the latter could remained unaligned. A variation to this heuristic is to iterate over the reference sentence instead, and for each reference word assign a link to its most similar output word. That way, we would obtain the opposite effect, all reference words would be aligned but not all output words. These two alternatives could be considered final solutions by themselves or they could be combined using one of the symmetrizations mentioned before. Once decided which is the final output-reference word alignment, we combine it with the input-output alignment to obtain an input-reference alignment that can be evaluated in terms of AER.

Table 3.2 shows the results of all these options: iterating over the output sentence (out-ref), iterating over the reference (ref-out), union, intersection and grow diagonal. We have used MGIZA and its force-alignment functionality to also align the same test set and compare the results. The force-alignment function allows the system to align unseen test data using existing models. For this experiment, we trained the alignment models with the same corpus used for the baseline system mentioned on subsection 3.4.2.1. MGIZA directly computes both input-to-reference and reference-to-input alignments and we simmetrized them with union, intersection and grow diagonal to explore the different options. Table 3.3 shows the evaluation results over the test set for the alignments obtained by MGIZA.

Results from both approaches vary depending on the heuristic used, resulting in better precision when intersection was used and better recall for the union. The best AER score can be found in the fourth column of Table 3.2, with the similarity function and the intersection symmetrization.

Chapter 4

Derived Units

Chapter 3 presented a translation-based word alignment strategy for an Augmented Corpus. The strategy was defined with the objective of producing an alignment that detects the errors made during decoding and their corrections. Now, we are going to explore different applications of that strategy, this time with the objective of improving the translation quality of a SMT system on a specific task. The plan is to augment the translation and the reordering models with new units that benefit the system.

We present different experiments in which the translation-based word alignment method proposed has proven to be useful. First, we will devote three sections to explain how to derive new translation units from these alignments and how to augment the baseline translation and reordering models with them. Then, we will present some experiments that put those ideas in practice. This experiments were conducted over systems using different SMT paradigms, language pair and domains, and they will help to better understand the rationale behind the algorithms presented so far.

The following list summarizes the experiments presented and the ideas they cover:

1. A Catalan-Spanish N-gram based SMT system (Section 4.4). This experiment was the first approximation to the Derived Units concept. Here we will cover the usability of change zones between highly similar languages. It is a special case of a domain adaptation task, where the objective is to keep the domain (news) but adapt it to new vocabulary and writing style. Results of this experiment showed statistically significant improvements over the baseline system with a 99% of confidence level and encouraged us to mature the concept of Derived Units and adapt it to other language pairs and paradigms. After this experimental result, when we tried to extrapolate to non-similar language pairs, we detected the necessity of introducing the similarity function described in Section 3.4.

2. An English-Spanish Phrase based SMT system (Sections 4.5.1 through 4.5.3). Most of the experimental work is done in this scenario. With a baseline system coming from the Europarl [52] and an adaptation corpus from the news domain, we jumped from change zones to the first version

of the similarity function presented in Section 3.4, where the features for unknown words were not yet designed. Nevertheless, the results showed again significant improvements with a similar confidence level as before. Within this task, we also tested the effects of increasing the Augmented Corpus size (until now, just a 0.5% of the baseline corpus) and its effects over the test set.

3. WMT 2012's English-Spanish Phrase-based SMT system (Section 4.5.4). We describe the system presented in the translation task of the Seventh Workshop on Statistical Machine Translation (WMT). In this experiment, the same strategy as in the previous experiment was used. However, in this case, a larger dataset was used and the analysis of unknown words was included along with the results of adding its two features in the similarity function.

4. Derived Units using post-editions and feedback provided by users (Sections 4.5.5 and 4.5.6): With these two experiments we move forward to applying the Derived Units strategy in Augmented Corpora built with real user feedback. The first experiment uses data from MT researchers, who built a post-edited collection of sentences for the Quality Estimation Task of the WMT 2012. The second one correspond to a real scenario, where the Augmented Corpus was collected from users of the Reverso.net web translator.

5. Derived Units over training corpus (Section 4.6). The last experiment explores a different application of the Derived Units concept. Instead of adapting a baseline system with additional material, we focused on improving the baseline material itself. Therefore, after completing the baseline system, we used it to translated the training corpus. We constructed the Augmented Corpus with those sentence pairs the system was not able to recover correctly. After that we produce a new word alignment and build a translation model which is used to build the improved system.

4.1 Building The Derived Model

After automatically computing the change zones or the word alignment from an Augmented Corpus, we can extract two types of translation units:

1. Previously seen translation units: these are units that the decoder used correctly during the translation phase and they produce the same solution as the reference. It is important to extract and identify them because they should be enhanced in the final model, so they can be chosen again if the system faces a similar situation.

2. New translation units: as a result of the change zones or the word alignment, we can extract new translation units that do not exist in the baseline translation model. These units are also important as they represent the main elements that can contribute to improve the decoder performance. We have to add them to the final translation model in order to provide the decoder with new options so it does not make the same mistake again when similar situations appear.

4.1.1 Derived Model For N-gram based SMT Systems

On N-gram based SMT systems, the translation model is represented by a bilingual language model built with tuples extracted from the training corpus. The procedure for building this model was explained in detail in Section 2.6.2. It involves vocabulary extraction, tuple filtering, unfolding (if reordering is considered) and finally building the language model.

With our Augmented Corpus segmented with change zones, we can build a translation model in the same way. As you can see at the bottom of Figure 3.4, the change zones provided a unique monotone segmentation that allows us to extract tuples in a straightforward manner.

This process also includes vocabulary extraction and filtering phases. Sometimes the post-edited version is very different from the translation output and the Levenshtein distance is not able to properly align the words to extract the correction tuples, resulting in noisy tuples that do not add value to the translation model. Going back to Figure 3.5 we can see an example of a free post-edition that causes more noise than information when it is compared to its output translation. To compensate this problem, we proposed a lexical filter to prune the tuple vocabulary, removing from it all tuples whose lexical cost is above a threshold.

Working with the word alignment instead of the change zones is also possible. The difference would be that we would have to build that monotonic segmentation beforehand and then extract the resulting tuples. Tuple extraction in this case would require the same procedure we use to build a baseline system from a word-based aligned parallel corpus.

4.1.2 Derived Model For Phrase-based SMT Systems

For the case of Phrase-based SMT system, we only use the word alignment as alternative for building the model. The change zones group all the related words in the same block, therefore it is not possible to extract smaller phrases than those provided by the zones. On the other hand, the word alignment strategy works just like the standard Phrase-based pipeline and, therefore, no extra work is needed to extract a translation model from it.

4.2 Model Combination

After computing the translation model with the augmented corpus, the next step is to combine it with the baseline system's model to obtain the final translation model. This combination follows the form:

$$TM(n) = \alpha TM_{baseline}(n) + (1-\alpha)TM_{post-edited}(n) \tag{4.1}$$

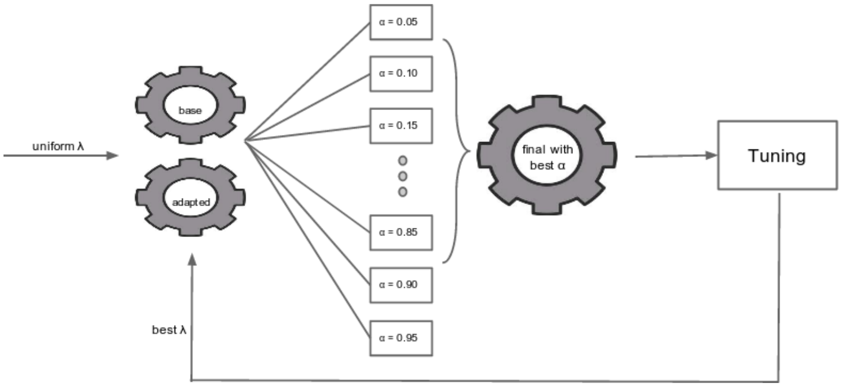


Figure 4.1: In order to compute α, we start with an initial set of values for λ_i and retrieve the value of α that gives the best performance over the development set in terms of BLEU; then, the tuning begins and updates the values of λ_i to iterate again until convergence of α or λ_i.

where $TM(n)$ is the final translation model score for translation unit n, $TM_{baseline}$ is the baseline translation model and $TM_{post-edited}$ is the new translation model recently obtained.

The process of combining the two translation models is tight with the tuning of the final SMT system. First, we will test the final system with different values of α and we will keep the one that performed better in terms of BLEU score. Then we start an optimization of the different model weights, using the value of α obtained before for interpolating both translation models. Once the optimization has ended, we test again the system with different values of α to see whether the optimal value for it has changed. If it has, then we start a new optimization process, beginning at the best point found during the last optimization. This process is repeated until no changes are detected in α or no improvements are observed in the BLEU score during the optimization. You can see a graphical representation of this procedure in Figure 4.1.

4.3 Augmenting the Reordering Model

Together with the translation model, the reordering model is highly tighted with the paradigm used for translation. In the case of Phrase-based Machine Translation, the dominant reordering is the one originally proposed by Tillman [76]. As it was seen in Section 2.7, this model includes different features depending on the movement made by the current phrase: monotone, swap or distant, as well as, whether we are considering source to target or target to source computations.

Moreover, this reordering model is lexical based, i.e., for every different pair there are seven different features to compute (the six implied above and the standard distance-based feature). Therefore, in order to obtain confident results, we would need a large corpus to obtain enough occurrences of every situation.

When we are working with an Augmented Corpus, this is not usually the case. If the Augmented Corpus comes from user's feedback, for instance, it is usually very small compared with the baseline corpus (around 0.5% of the baseline). In case of domain adaptation, the scenario is similar although not that limited.

For that reason, we proposed to augment the reordering model in the following manner:

1. Compute the reordering model of the Augmented Corpus considering all phrases extracted, i.e., just as if we were building a new system.
2. For all new phrases that were extracted from the Augmented Corpus and do not exist in the baseline reordering model, include their computed features in the baseline model as new entries in the table.
3. For all phrases that were extracted from the Augmented Corpus but do exist in the baseline model, do not change their reordering features. Instead, keep the values computed with the baseline model.

Following these three steps, we are conserving the baseline reordering model and just augmenting it with the new phrases found in the Augmented Corpus.

4.4 Preliminary Steps With a N-gram based SMT

Our first approach to the derived units methodology was tested on a Catalan-to-Spanish N-gram based SMT system. The objective was to adapt an already tuned translator, trained with a corpus collected from old news, with a small additional corpus collected from more recent news. We did not plan to change the news domain but adapt it to the current time, adding new vocabulary and adapting the writing style.

All corpora used for this experiment came from news domain. The baseline training corpus was a Catalan-Spanish parallel corpus collected from the local newspaper “El Periódico” during the period 2000-2007 while the corpus used for adaptation, came from the same newspaper but it was collected from 2008 news.

Statistics of the baseline and adaptation corpora can be found in Table 4.1. Detailed information about these corpora can be found in Appendix A.1. For tuning, we used the same development corpus used in the baseline system.

For this experiment, the Augmented Corpus was built translating the source side of the adaptation parallel corpus. Therefore, the derivation process is based on the comparison between the source side translation provided by the baseline system and the target side of the corpus.

As for the baseline system, we used N-II, a freely available online translation system for Catalan and Spanish whose current version is described in [31].

The experiment proceeded as follows:

	Training		**Derived Units corpus**		**Test**	
	Catalan	Spanish	Catalan	Spanish	Catalan	Spanish
Number of sentences	$4.6M$		1608		2048	
Running words	$96.94M$	$96.86M$	$34.67K$	$35.17K$	$46.03K$	$46.00K$
Avg. words per sentence	21.07	21.05	21.56	21.87	22.47	22.46
Vocabulary	$1.28M$	$1.23M$	$11.02K$	$11.17K$	$13.28K$	$13.39K$

Table 4.1: Statistics of the corpora used with N-gram-based SMT experiments

System	Tuples
Baseline	$1.16M$

Filter	Extracted	Unseen
None	$8,511$	$1,360$
Lexical	$8,307$	$1,097$

Table 4.2: Extracted and unseen tuples

1. We translated the source side of the 2008 corpus to build the Augmented Corpus.

2. Derived Units were extracted following the procedure described in Section 4.1.1, using change zones.

3. A small translation model was built.

4. This small translation model coming from the Augmented Corpus was combined with the baseline system to produce the final translation model.

5. The system was evaluated with the test corpus described in Table 4.1 and compared with the baseline system.

Because the Catalan-Spanish baseline system did not have a reordering model, we did not compute it in the Augmented Corpus either.

The third and fourth steps include a lexical filter and the value of α for model combination. As explained in Section 4.1.1, the lexical filter help us remove noisy tuples from the tuples vocabulary, so they do not appear in the translation model. Table 4.2 shows the number of tuples initially in the baseline system, the ones extracted from the Augmented Corpus and the final number of tuples after the filtering. It also shows how many of those extracted tuples did not appear in the baseline system. It can be seen that 20% of the new tuples were removed after the filtering.

In order to determine the value of α for model combination in the fourth step, we followed a simplified form of the strategy proposed in Section 4.2. We just tuned different systems, each of them built with a different value of α and we kept the one that performed better in terms of BLEU score after one optimization. The optimum value for α was found to be $\alpha = 0.85$ and therefore it was the value used in the rest of the experiment. Figure 4.2 and Table 4.3 show the variation of the BLEU score across the different values of α and its maximum in $\alpha = 0.85$.

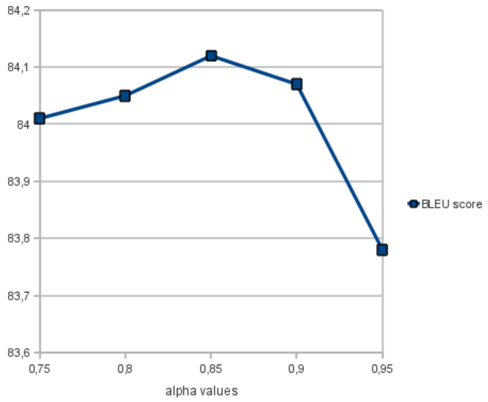


Figure 4.2: BLEU variation during tuning using different values of α

α	0.75	0.80	0.85	0.90	0.95
BLEU	84.01	84.05	84.12	84.07	83.78

Table 4.3: α-values used to combine the translation models and its corresponding BLEU score after tuning

We built two different systems, in order to evaluate the effect of the tuple filtering. The results obtained over the correction and test corpora can be seen in Table 4.4. The second and third column correspond to the BLEU scores obtained by the different systems in the correction and test corpus, using only one reference. The fourth column gives the confidence level for BLEU scores over the test set with respect to the baseline.

Notice that all augmented system performed better in the correction corpus, which is expected because it is part of the Augmented Corpus. Also, they all improved the baseline test score. It is interesting to notice that once we added the lexical filter, the correction BLEU decreased and the test BLEU increased. It means that the filter helped the system to generalize its learning.

Moreover, even though the augmented system without filter is not significantly better than the baseline, we found that with the lexical filter we achieved a better performance, with a confidence level of 99%. The confidence levels were obtained using the “Pair Bootstrap Resampling”' method described in [51].

Besides the automatic test described before, we also computed the BLEU scores over the test set, sentence by sentence, with the baseline and the final system; then, we compared them to determine which sentence translations were better (or worse) in the final system and why. Table 4.5 shows these results. We can see that the final system was better in 154 sentences, the baseline system was better

System	Correction	Test	Confidence Level
Baseline	76.87	77.19	-
Derived tuples w/o filtering	85.96	77.23	87.24%
Derived tuples with filtering	83.85	77.33	99.20%

Table 4.4: Results of comparing the tuple filtering process.

System	Higher	Lower	Same
Baseline	122	154	$1,722$
Derived tuples with filtering	154	122	$1,722$

Table 4.5: Number of sentences that have a higher, lower or same BLEU score between the baseline and the derived system.

in 122 and that they got the same score in the remaining 1772. We took a closer look at those 122 sentences and found that most of the final system outputs had used synonyms and paraphrases and that they were indeed valid although they were not used in the reference. On the other hand, we found some semantic, lexical and morphological errors solved among the 154 sentences where the final system had a better score.

Figure 4.3 shows a sample of both subsets, displaying first the baseline output and then the final system output. The first case corrected a word-by-word translation; it means "besides". The second and sixth pair are example of sentences with unknown words, "ciríl·lic" and "Govern", that are solved with their correct translation by the final system, "cirílico" and "Gobierno Catalán"; their English translation are "Cyrillic" and "Catalan Government". The third one was the catalan word "drets" that has two meanings and the wrong one, "de pie", was chosen by the baseline; "de pie" stands for "stood (up)" while "derecho" means "right", like in "civil right". The fourth final system output corrected a morphological error, choosing the verb with the proper person and number. Pair number five presents two synonyms. The seventh pair adds a preposition that could also be omited, as the baseline output did. Finally, the last pair presents two different ways of saying the same ("on the other hand") and they are equally valid even though the BLEU score is lower in the final system because the reference matches the baseline.

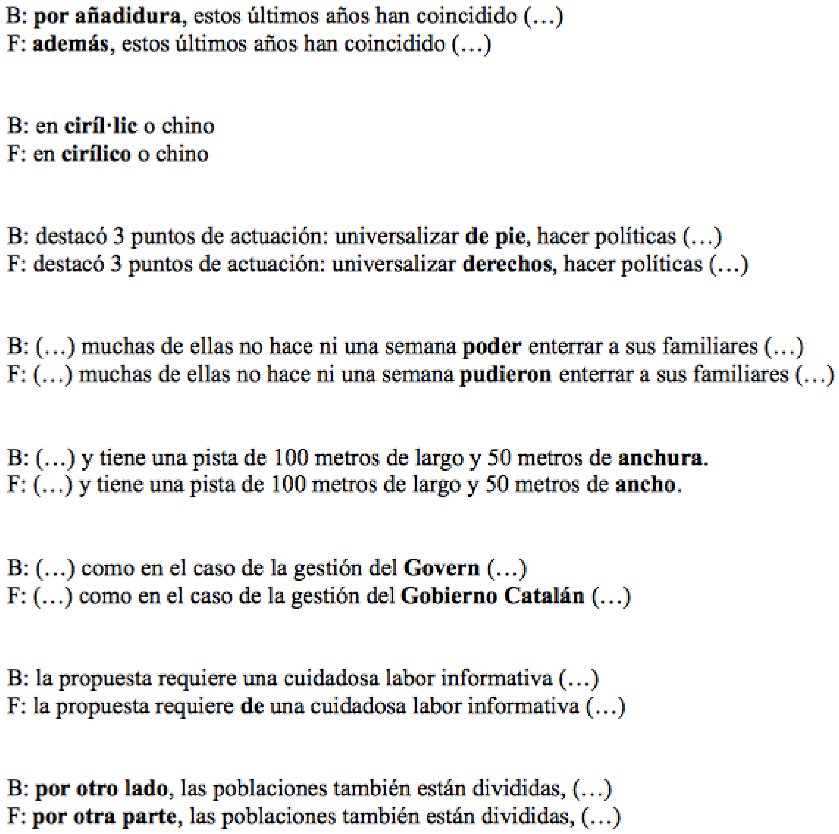
The results from this experiment show that the Derived Units are a valid strategy to improve SMT system quality in terms of BLEU score. Therefore we felt confident to go further and evolve the concept applied here to a different language pair and a different translation paradigm.

4.5 Domain Adaptation

One of the most active research areas in MT during the last years has been domain adaptation. It consists in adapting a general system (commonly refered to as out-of-domain) to a specific domain using either monolingual or parallel data from the domain we want to adapt to.

In these experiments, we used the translation-based word alignment presented in Chapter 3 together with the concept of Derived Units introduced in this chapter as a valid strategy for domain adaptation.

These experiments were proposed as an extension of the idea of Derived Units explored in Section 4.4. They build upon the concepts worked before but differ from those in several aspects:



B: **por añadidura**, estos últimos años han coincidido (…)
F: **además**, estos últimos años han coincidido (…)

B: en **ciríl·lic** o chino
F: en **cirílico** o chino

B: destacó 3 puntos de actuación: universalizar **de pie**, hacer políticas (…)
F: destacó 3 puntos de actuación: universalizar **derechos**, hacer políticas (…)

B: (…) muchas de ellas no hace ni una semana **poder** enterrar a sus familiares (…)
F: (…) muchas de ellas no hace ni una semana **pudieron** enterrar a sus familiares (…)

B: (…) y tiene una pista de 100 metros de largo y 50 metros de **anchura**.
F: (…) y tiene una pista de 100 metros de largo y 50 metros de **ancho**.

B: (…) como en el caso de la gestión del **Govern** (…)
F: (…) como en el caso de la gestión del **Gobierno Catalán** (…)

B: la propuesta requiere una cuidadosa labor informativa (…)
F: la propuesta requiere **de** una cuidadosa labor informativa (…)

B: **por otro lado**, las poblaciones también están divididas, (…)
F: **por otra parte**, las poblaciones también están divididas, (…)

Figure 4.3: Output samples from the baseline and final systems. Every pair presents first the baseline output (labeled “B”’) and then the final system output (labeled “F”). The first four pairs are examples of a higher final BLEU, the last four pairs had a higher baseline BLEU. Differences between translations are higlighted in boldface.

- They are based on an English to Spanish SMT system instead of Catalan to Spanish: this change of language pairs represents a generalization of the unit extraction algorithm because it introduces the concept of reordering (not handled before).
- We worked with Phrase-based SMT system instead of N-gram based: Phrase-based SMT SMT systems are not as restrictive as N-gram based in terms of translation units. Therefore, working only with change zones would limit the number of possible phrases.
- Different Augmented Corpus sizes were considered: we explored the effects of the Derived Units as the Augmented Corpus size grows up to 16 times its original size.
- The derived corpus method is compared with other state-of-the-art domain adaptation techniques.
- We explore the use of user feedback as a special case of domain adaptation.

We have divided the description of these experiment in subsections because of its length. First, we describe the initial set-up, including the baseline system, the different domain corpora and the first results of a small Augmented Corpus; then, we compare those results with three different alternatives for domain adapation; finally, we describe the effects of increasing the Augmented Corpus size.

Except for the experiment mentioned in Section 4.5.6, which is based in web users, all other experiments described in this section use the same scenario: adapting a Phrase-based system trained with sessions from the Europarl to a news domain. This domain adaptation task was first proposed in the Second Workshop on Statistical Machine Translation, in 2007 [14].

4.5.1 Initial set-up and first results

The baseline system was built using only the Europarl corpus version 5 (Appendix A.2) as training material for the translation model. Development and testing used news domain corpus from WMT 2010 (Appendix A.3).

The system built was a standard Phrase-based SMT system with two language models: one for surface words and an additional one for Part-Of-Speech (POS). Preprocessing of the data included lowercasing and tokenization.

Besides the Europarl corpus for training, we used additional monolingual corpora to build three language models for the target. Each language model correspond to a different domain: Europarl, News and United Nations (Appendix A.4); and the final language model is the interpolation of the three. The interpolation process aimed to minimize the perplexity over the development corpus. The POS language model was computed using only the Europarl corpus.

Apart from the standard training corpus used to build the baseline system, an Augmented Corpus was needed to perform the derivation process. We considered the News parallel corpus available for WMT

	Sents.	Words	Avg.	Vocab.
Src. (ES)	1500	42K	27.86	7.7K
Trans. (EN)	1500	39K	25.67	6.3K
Corr. (EN)	1500	36K	23.90	6.9K

Table 4.6: Augmented Corpus statistics for source (Src.), translation output (Trans.) and correction corpora (Corr.)

Corpus	Baseline	Correction
Total Phrases	96M	111K
Used in Dev.	4.0M	15K
New in Dev.	-	5K
Used in Test	5.5M	19K
New in Test	-	6K

Table 4.7: Phrase table statistics

2010 as our starting point because the test set was from the news domain. From this training corpus, we randomly extracted 1500 sentences and translate its source side using the baseline system. We chose 1500 sentences to make this experiment comparable with the one presented before on Catalan-to-Spanish. At the end, the Augmented Corpus consisted of the source and target side, and the translation output. The target side was considered the correction corpus. Table 4.6 shows statistics for the used corpora.

Once we had defined the correction corpus, the process followed the steps describe in Sections 4.1, 4.2 and 4.3. Table 4.7 shows the number of phrases useful for tuning and testing that we had in our baseline system along with the ones resulting from the Derived Units.

In our first experiment, we evaluated our method using the internal test set from WMT 2010. For this case, we used the derived TM, which were interpolated with the baseline TM as described in Section 4.2, as well as the unseen reordering phrases, which were added to baseline reordering model. Four different tunings were performed to consider the effect of MERT's random initialization.

We compared our approach with the naïve domain adaptation strategy consisting of concatenating the out-of-domain corpus with the in-domain corpus to build from scratch a new SMT system. In this case we concatenated the Europarl Corpus with the 1500 in-domain sentences.

System	min.	max.	avg.	p-value
Baseline	22.52	22.83	22.67	-
Concatenation	22.55	22.72	22.64	0.401
Derived	22.88	23.08	22.95	0.001

Table 4.8: System results and significance of its difference with the baseline system. The first column shows the lowest BLEU score achieved after four independent tuning processes, the second columns shows the highest and the third one the average of all four tuning phases.

Augmented Corpus size	min.	max.	avg.
0 (Baseline)	**22.52**	**22.83**	**22.67**
1500	**22.88**	**23.08**	**22.95**
3000	22.87	23.04	22.95
6000	22.71	23.09	22.98
12000	22.91	22.99	22.95
24000	22.95	23.17	23.04
24000 (Concatenation)	**22.88**	**23.02**	**22.92**

Table 4.9: System results using incremental augmented corpora

Table 4.8 shows the result of all system configurations. The Derived Units approach is the strategy that obtained the highest BLEU score and its difference is statistically significant when compared with the baseline system. Also, we can notice that the concatenation approach did not benefit the translation quality as compared to the baseline.

4.5.2 Increasing the Augmented Corpus size

The previous result showed that our proposed methodology is a valid domain adaptation strategy when the Augmented Corpus is small compared to the training corpus. Until now, we have worked with an Augmented Corpus that represents 0.5% of the training material. The following results (shown in Table 4.9) describe the evolution of the system performance as the augmented copus size is incremented. Starting with the initial 1500 sentences Augmented Corpus, we adapted the system using different Augmented Corpora of size 3000, 6000, 12000 and 24000. All these systems share the same language models and development corpus. The only component that varies among them is the augmented corpora size, which is incremental, i.e., the 3000 sentences Augmented Corpus contains all sentences from the 1500 sentences Augmented Corpus; the 6000 sentences contains all sentences from the 3000 and so on. The last row of the table is the result of concatenating the 24000 sentences Augmented Corpus to the baseline training corpus.

From Table 4.9 we can extract some interesting conclusions:

1. The Derived Units are able to extract valuable information from small augmented corpora.

2. The size of the Augmented Corpus is not proportional to the information it provides to benefit the system if it represents less than 1.5% of the training material.

3. Compared to the concatenation strategy, the Derived Units were able to provide the same benefits to the final system using only 1/16 of the in-domain data. Using the same 1/16, the concatenation strategy performed just like the baseline system, as we saw in Table 4.8.

4.5.3 Comparing to other domain adaptation alternatives.

Besides concatenating corpora to adapt the out-of-domain corpus, other alternatives to domain adaptation are available. Moses, for instance, allows working with two translation models at the same time. Using this feature we could work directly with the baseline translation model and the derived translation model and let the optimization handle the weights for each feature.

When using multiple translation models, we need to specify how the tables will work during decoding. Moses offers two options[1]:

- Both translation tables are used during decoding: it means that each phrase is scored by each table and all scores are used during decoding. This mode needs that all phrases appear in both tables, otherwise the phrases will not be scored.
- Either translation table is used during decoding: with this configuration, the translation scores are collected from one table or the other. In case there is a phrase that appears in both tables, two different translation path are created, each of them with a different set of scores, corresponding to each table.

Apart from the alternatives offered by Moses, another method to perform domain adaptation is to obtain the word alignment of the Augmented Corpus using a different strategy. MGIZA is a tool that was initially built to perform the word alignment in a multi-thread enviroment but it also include an interesting additional feature called “forced-alignment”. Using the forced-alignment functionality of MGIZA, we are able to use previously trained alignment models to align new sentences on demand. For domain adaptation, we could align the Augmented Corpus directly with the models learnt during the baseline training and complete the adaptation of “Derived Units” using that alignment set instead of the one obtained with our translation-based alignment method.

In order to compare our Derived Units strategy for domain adaptation with the methods mentioned before, we built three new systems, each of them with a different domain adaptation strategy and we contrast their translation results in terms of BLEU with our previous system built with an Augmented Corpus of 1500 sentences. The results can be seen in Table 4.10. Again, the strategy based on translation-based word alignment outperformed the other domain adaptation methods.

4.5.4 Deriving Units for domain adaptation in an international evaluation campaign.

Given the positive results we achieved with the previous experiments, in 2012 we decided to present the Derived Units strategy into the Workshop of Statistical Machine Translation; specifically to the

[1] http://www.statmt.org/moses/?n=Moses.AdvancedFeatures#ntoc18

Domain adaptation strategy	BLEU
Baseline	**22.67**
Derived Units method	**22.95**
Multiple tables “both”	20.54
Multiple tables “either”	22.75
MGIZA with the 1500 corpus	22.87

Table 4.10: System results using incremental augmented corpora

Corpus		Sentences	Words	Vocabulary	Average length
Development	English	7.57K	189.01K	18.61K	24.98
	Spanish		202.80K	21.75K	26.80
Internal test	English	3.00K	74.73K	10.82K	24.88
	Spanish		81.01K	12.16K	26.98
Official test	English	3.00K	72.91K	10.24K	24.28
	Spanish		80.38K	12.02K	26.77

Table 4.11: Details of the development and test corpora used to train the systems

Translation’s Task. This participation allowed us to compare the methodology with the current state-of-the-art strategies for domain adaptation.

Following the same framework as before, the training material for this task started with the Europarl and its testing domain was again, news commentaries. The corpus, however, was different, as it is updated every year with new information. For WMT 2012, the Europarl corpus corresponded to version 7 (Appendix A.2) and the development and test data from News were new sets (Appendix A.3). Table 4.11 shows the statistics for the development and test corpora while the statistics for the training material can be found in the Appendixes.

Additionally, to build the language model of the baseline system we used monolingual corpora from the Europarl (Appendix A.2), monolingual corpora from News Commentary (Appendix A.3), the Gigaword monolingual corpus in English and the United Nations parallel corpus (Appendix A.4).

Aiming at building a system with all the available data, the settings for these experiments were as follows:

- The baseline system was trained with the Europarl and United Nations corpora.
- The final language model was built from multiple language models, each of them from an independent corpus (Europarl, UN, the parallel News Commentary and each monolingual news corpus). We combined them in a final language model of minimal perplexity over the development corpus. This language model was the same for all compared systems.
- We used all the parallel News Commentary corpora to build an Augmented Corpus and applied the Derived Units domain adaptation strategy.

		BLEU	NIST	TER	METEOR
Internal test	**Baseline**	**33.47**	**8.19**	**49.88**	**52.10**
	Derived Units	**33.90**	**8.25**	**49.37**	**52.34**
	Multiple tables “either”	33.79	8.23	49.51	52.18
	Multiple tables “both”	32.49	8.09	50.55	51.25
	Incremental training	33.15	8.13	50.21	51.76
Official test	**Baseline**	**31.73**	**8.02**	**50.82**	**51.42**
	Derived Units	**32.13**	**8.07**	**50.51**	**51.81**

Table 4.12: Results from WMT 2012’s Translation Task

- The translation-based word alignment included for the first time the features related with unknown words.

For this experiment, we computed multiple automatic evaluation metrics using the Asiya [39]: BLEU, NIST, TER and METEOR; and we performed a comparison between our approach and the multiple phrase table approaches from Moses. We also compared our approach with another alternative for domain adaptation called: incremental training.

Incremental training is another functionality included in the Moses framework that allow us to update the word alignment models and append new sentences to the corpora along with their alignments in order to produce a new translation model on demand, instead of concatenating it all in a batch process. Incremental training is similar to the force alignment described before in the sense that it starts from trained material. However, with incremental training all models are updated at the end while force alignment just uses the old models to align new material, without modifying the former.

Table 4.12 shows the results for all systems mentioned before. The comparison with the other domain adaptation techniques was only done with the internal test. As we can see, the deriving units strategy improves the baseline system 0.43 BLEU points in the internal test and 0.4 points in the official test and also improves the score of the other three metrics. Regardless the other domain adaptation alternatives, we found that the system named Multiple tables “either” performed better than the baseline by 0.32 BLEU points. The difference between the Derived Units and baseline scores in terms of BLEU has a confidence level of 95%. A comparison between our systems and the rest of participants during the Translation’s Task can be seen in [15].

4.5.5 Post-editions from an international evaluation campaign as a special case of Domain Adaptation

The motivation for the Derived Units strategy comes from user feedback. The idea was to design a method that allows a system to correct its errors automatically using feedback collected from users. This experiment shows the benefit of the method in such a case.

	Sents.	Words	Avg.	Vocab.
Src. (EN)	1800	53K	29.24	8.5K
Trans. (ES)	1800	55K	30.41	9.2K
Corr. (ES)	1800	57K	31.15	10.2K

Table 4.13: Augmented corpus statistics for source (Src.), translation output (Trans.) and correction corpora (Corr.) used on the feedback experiment

	BLEU	NIST	TER	METEOR
Baseline	25.04	7.09	57.48	50.05
Derived Units	25.75	7.18	56.86	50.41

Table 4.14: Results for the feedback experiments

Until now, we had built the Augmented Corpus using in-domain parallel data. The three components of the Augmented Corpus were the source side, its translation provided by the baseline system and the target side as a correction for the system.

For this experiment we built an Augmented Corpus using a set of post-edited sentences collected from human experts. Therefore the Augmented Corpus included the input sentences, its translation and the post-edited version of the latter provided by human editors.

Similar to the previous experiments, the baseline was built using the Europarl and News Commentary corpora. The feedback corpus came from a set of 1800 sentences that were automatically translated and manually post-edited. Appendix A.5 shows the statistics for the baseline system (training, development and test) and Table 4.13 shows the statistics for the augmented corpora. The baseline system and post-edited material correspond to those used during the Quality Estimation Task of the WMT 2012 [15].

As we can see in Table 4.14, the Derived Units produced an increment of 0.7 BLEU points in the test data and also improved all other automatic metrics. As before, the differences between the Derived Units and baseline scores in terms of BLEU have a confidence level of 95%.

4.5.6 User Feedback collected from web users

The previous experiment evaluated the proposed strategy with real feedback, i.e. the Augmented Corpus was built using post-editions of output translation instead of simulating feedback with a target reference. However, this feedback came from researchers of machine translation, who tried to write post-editions that were as close as possible to the original translation in order to learn new features for the 2012's WMT evaluation metrics campaign. Therefore, all data provided could be safely used to improve the SMT system, just like the target references in the previous examples.

This section describes a more realistic scenario, extracted from the results of Barrón Cedeño et. al. [7], in which the system was available to the general public through a website and users from all over the

	BLEU	NIST	TER	METEOR
FAUST Raw				
Baseline	34.47	8.28	51.76	54.87
Feedback filtered	**35.22†**	**8.41***	**50.19**	**55.83**
Feedback w/o filter	34.41	8.27	51.15	55.32
FAUST Clean				
Baseline	38.64	8.68	46.91	58.42
Feedback filtered	**3**9.49†	**8.80***	**45.42**	**59.31**
Feedback w/o filter	38.61	8.64	46.80	58.60

Table 4.15: Results for the feedback experiments. '$*$' and '†' indicate confidence levels of 0.99 and 0.95, respectively

world translated sentences of any domain and provided feedback as they wished, producing a very noisy collection of data [69]. The paper proposes a Support Vector Machine (SVM) classifier for feedback filtering and we collaborate in the research applying the Derived Unit strategy with the filtered data to evaluate the effect during translation.

The filtering was applied over a collection of $6.6K$ user feedback instances, provided by Reverso[2] from its MT weblogs. After receiving the filtered feedback, we used it to build an Augmented Corpus to enhance the translation model of the basleline Phrase-based SMT system by using the Derived Units strategy. For the alpha parameter of the Derived Units strategy (used to weight the contribution of the baseline and derived phrase tables) we chose $\alpha = 0.60$ from the experience of the previous experiments. Also, similarly to the previous experiments, the language model was the result of a combination of language models from news, UN and Europarl, with the addition of a new language model build with monolingual data provided by Reverso. The development set used was the FAUST clean corpus. The systems were evaluated with two test sets: the raw and clean input of the FAUST test corpus and its corresponding translation as reference. Details on the Augmented Corpus, the monolingual data for the language model and the FAUST corpus used for development and testing can be seen in Appendix A.5.

Table 4.15 presents the results on the FAUST test set for three system configurations: the baseline system (Baseline), a derived system using the filtered feedback and a derived system using all the feedback, i.e. without the filtering proposed in [7]. The automatic metrics used to measure the performance of the different systems were BLEU, NIST, METEOR and TER. In the table, the symbols '$*$' and '†' indicate confidence levels of 0.99 and 0.95, respectively).

From the results obtained in this experiment, we observe that introducing feedback without any filtering (the rows "Feedback w/o filter" in Table 4.15) does not improve the MT system performance. Thus, the feedback cannot be added as such with unfiltered noise. In the previous experiment we were confident in using all feedback (and obtained significant improvement in translation quality) because it was either simulated with translation references or collected from post-editions made by human experts

[2] http://www.reverso.net/

for research purposes. From these results we can conclude that the feedback approach proposed by Barrón Cedeño et. al. [7] together with the Derived Units strategy described in this thesis introduce a significant improvement over the baseline system, providing a learning methodology well-suited to the problem of user feedback.

Qualitative Analysis of the Adaptated System

Additional to the automatic results reported in [7], we will now present an human analysis focused on the different phenomena that the system was able to fix and adapt.

The analysis was done by five experts who studied 414 translation-triplets from the 50% selected feedback (FAUST Clean) including: source language sentence, baseline translation and the translation made with the improved system. For each of the sentences, the annotators were asked to compare the "baseline" and "improved" translations. The possible outputs for the comparison were "better" (if the "improved" output was better than the baseline), "worse" (in case of the opposite), "same" (if the change did not alter the general quality) and ambiguous / can't say (if it was not possible to determine whether the changed was for better or worse).

The comparison was done at two levels:

1. Overall adequacy and fluency.
2. Detailed level identifying change at different phenomena.
   (a) Function words: it includes insertion, replacement or deletion of function words.
   (b) Word fertility: it includes insertion, replacement or deletion of non-function words.
   (c) Lexical changes: it includes choosing a different translation for the same source phrase and Out-Of-Vocabulary (OOV) translations that the baseline could not translate.
   (d) Reordering.
   (e) Morphology: it includes changes in person, genre, number and tense of verbs and nouns.
   (f) Harmful element (eg, a mistranslation caused by a bad feedback) into the model.

Finally, we selected 10 common translation-triplets for all annotators in order to compute the agreement according to Cohen's kappa, resulting in $\varkappa = 0.57$. Results are presented in Table 4.16.

Once we had analyzed the qualitative analysis results, we observed lexical translation to be the main aspect that feedback usage is affecting ($\approx$ 60% Better+Same+Worse). These changes mostly came from corrections based on mistranslations other than minimizing the impact of OOV, as they were reduced only by 0.3%. Other significant aspects that are affected by our methodology are reordering (22%) and morphology (20%). In a third term would be the function words (14%) and the insertion/deletion

	Better	**Same**	**Worse**	**Can't Say**
Adequacy	34.54%	40.58%	15.70%	9.18%
Fluency	32.61%	49.76%	17.63%	-

	Better	**Same**	**Worse**	**Can't Say**
Function Words	6.52%	2.42%	4.10%	86.96%
Fertility	5.56%	3.62%	8.45%	82.37%
Lexical	28.74%	19.32%	12.32%	39.62%
Reordering	12.08%	3.86%	5.07%	78.99%
Bad Feedback	-	-	5.31%	94.69%
Morphology	7.00%	7.97%	4.59%	80.44%

Table 4.16: Results of the qualitative analysis of 414 sentences by 5 human annotators

of words. However, concerning word fertility, we see that the feedback changes worsen the system performance (8.45%) rather than improving it (5.56%). At last, we observed that only 5% of the changes were due to bad feedback learning (noise not filtered).

These results confirm the intuition that our proposed feedback-based learning strategy allows to incrementally improve a translation system beyond the simple lexical coverage and disambiguation, as it corrects also other phenomena such as reordering and morphology.

4.6 Derived Units Over Training Corpus

As a last experiment, we proposed a different application for the translation-based word alignment and Derived Units strategy. In this case, the rationale is as follows: if we were able to improve an out-of-domain system using an in-domain Augmented Corpus to correct the errors the decoder made in a different domain, we could built an Augmented Corpus using the training material in order to correct the baseline system itself.

This approach is related to the work of Wuebker et. al. [84], where the authors proposed a leaving-one-out strategy to train a translation model using a forced alignment of the training data obtained from the baseline system. This forced alignment is later used to estimate new phrase models that are combined with the baseline models to built a new system.

The difference between our proposal and theirs is that we used the trained system without constrains to translate the source side of the training corpus and then we compared the output with its target side in orden to obtain an alignment between source and target. Wuebker et. al. translated the source side using the target as a guideline to obtained a forced alignment directly from the decoder. Despite of this difference, the idea of using the baseline system to produce a new alignment for the training data in order to compute new phrase models is common in both works.

For this experiment, we used version 5 of the Europarl (Appendix A.2) to build a Spanish to English SMT system. For development and testing we used the Europarl corpus as well, because the goal of

	BLEU
Training	58.83
Development	32.76
Test	32.46

Table 4.17: Baseline BLEU's scores

	Baseline phrase-table
Total	$96,864,426$

	Derived phrase-table
Total	$91,937,387$
Known	$65,144,491$
New	$26,792,896$

Table 4.18: Phrase-tables' Statistics

this experiment is not domain adaptation but to exploit the training material as an Augmented Corpus itself.

The performance of the baseline system in terms of BLEU score over the development and test sets can be seen on Table 4.17. Because we needed the source side translation to extract the Derived Units, Table 4.17 also shows the BLEU score measured over the training set.

The experimentation started following the procedure described in Section 4.5: using the Augmented Corpus built with the training material, we built a translation-based word alignment for it and we used this alignment to compute a new phrase table and reordering model. Table 4.18 shows the statistics of the resulting phrase table (named derived phrase-table) compared to the baseline phrase table. The rows "Known" and "New" show the number of phrases extracted from the Augmented Corpus that did and did not exist in the baseline phrase table, respectively. We can see that the resulting phrase-table is 5% smaller and it includes 67% of the baseline phrases. The ratio between New and Known phrases in the new table is aproximately 3 : 7.

In this experiment we did not combine the derived phrase-table with the baseline phrase-table using the α weight of Section 4.2. Instead, we built the system using the derived phrase-table directly (and its corresponding reordering table). After tuning the model's weights with the development corpus, we translated the test corpus and compare the results with the baseline system. Table 4.19 shows the BLEU score over the development and test corpus for the final system.

It can be seen that translation of the test set benefited from the Derived Units strategy. An additional

	Derived Units
Development	32.63
Test	**32.76**†

Table 4.19: BLEU scores with the Derived Units phrase-table. The symbol "†" indicates a confidence level of 95% with respect to the baseline result.

test between the baseline and the derived system was made using the Paired Bootstrap Resampling Method [51] and the result was that the latter performed better in terms of BLEU score with a confidence level of 95%.

In conclusion, we have applied the Derived Units strategy to improve an SMT system within its own domain and without using additional data. The experiment consisted of translating the source side of the training corpus to use the translation output as pivot in order to build a new alignment for the training parallel data. The objective of this alignment, similar to the previous experiments, was to reinforce the good translation achieved by the baseline system and correct the sentence pairs that it could not reproduce. The main difference between this scenario and the original domain adaptation approach is that we did not combined the augmented translation model with the baseline model but use the former directly as our final model. Results showed that the augmented system performed better than the baseline with a confidence level of 95%. From these results we can conclude that the alignment provided by the translation-based word alignment is not only usefull for domain adaptation, but also for improving a system within its own domain.

Chapter 5

Conclusions

This dissertation has been focused on describing new strategies to benefit from user's feedback to automatically improve a Statistical Machine Translation System. We began with an overview of the state-of-the-art in SMT. Then, we presented the problem of domain adaptation, a particular research area of SMT. After describing the domain adaptation problem we introduced the reader to the user's feedback problem as a specific case of domain adaptation.

In the problem of user's feedback, we have a SMT system which is freely available on the internet. Users of the system provide us with translation alternatives for the sentences the system did not translate properly. Using the sentences the users wanted to translate, the translation provided by the SMT system and the alternative translation given by the users, we were able to design a method to improve the translation quality as measured by both automatic and human metrics.

The contributions made to the SMT field that can be found in this Ph.D. thesis are:

- We introduced the concept of Augmented Corpus, a collection of billingual parallel documents, where each document is a triplet consisting of the source side of a sentence, an automatic translation of the source side provided by a machine translation system and the target side, which is a valid translation (either from user feedback or a reference translation) for the source side.

- We defined a similarity function that compares the automatic translation and the reference of an Augmented Corpus to detect which zones of the translation were changed by the user to improve the automatic output. The results of these associations are then concatenated with the standard word alignment provided by the system during translation. As a result, we obtain a word alignment between the input and the feedback. The similarity function is based on a linear combination of different features that model how related a word from the automatic translation and a word from its corresponding feedback are.

- To obtain full benefits of the similarity function, the contribution weights for the different features must be tuned using a manually aligned corpus. We showed how this tuning can be done and how the AER is reduced during this phase. We obtained the contribution weights using a manually aligned set from the Europarl corpus and we observed translation improvements in all our tested domains: news commentaries, web translation queried from real users and the Europarl.

- We compared the alignment strategy derived from using the similarity function with a state-of-the-art alignment tool (MGIZA). The comparison was conducted over five different configurations, including three symmetrizations to obtain the final alignment. Results showed that, in terms of AER, our proposed strategy performs approximately as well as MGIZA and does not need the alignment model to be implemented.

- To tackle the user's feedback problem, we defined a domain adaptation methodology based on our alignment strategy. It consisted of four different steps: first, we built an Augmented Corpus gathering user's feedback from real translation; second, we aligned the input sentences with the feedback sentences using our alignment strategy and similarity function; third, we used this new word aligned corpus to extract translation and reordering units using the standard methods of extraction and scoring for Phrase-based SMT; fourth, we combined the baseline translation model and augment the baseline reordering model with the units derived in the previous step to build our final SMT system.

- We tested our proposed method in different scenarios following a progressive path. First we tested it in a Catalan-Spanish N-gram based SMT system to observe its performance with highly similar languages. Then, we studied the English-Spanish pair with a Phrase-based SMT system using an Augmented Corpus built with artificial feedback, i.e., we considered the reference translations of the parallel corpus as if they were user's provided feedback. This experiment allowed us to see if the domain adaptation strategy was able to learn from the decoder mistakes. Once we observed that an improvement in translation quality was achieved, we proceeded with a test using real user's feedback.

- The strategy proposed in this work achieves significant improvements in terms of BLEU score even if the Augmented Corpus represents only a 0.5% of the training material.

- Results from our evaluations also indicate that the improvement achieved with the domain adaptation strategy is measurable by both automatic a human-based evaluation metrics.

Appendix A

Corpora Description

A.1 El Periódico

This corpus is a news domain Catalan-Spanish corpus collected from the bilingual newspaper "El Periódico" covering news from 2000 to 2007. The most valuable characteristic of the newspaper in terms of bilingual resource, is that its news in both editions have the same text but in different language, i.e. they are literal translation of each other, making it a easy to build sentence-level parallel corpus as opposed to other multilingual media where the parallelism is only document-based.

It has more than 4 million sentences and over 95 million running words both in Catalan and Spanish with a vocabulary of over one million words. Its main statistics can be seen in Table A.1.

Additional to the 2000-2007 period, more recent news were extracted from 2008, collecting 150 thousands more sentences of parallel data. The statistics for this segment can be seen in Table A.2

A.2 Europarl Corpus

The Europarl Corpus is extracted from the sessions of the European Parliament. It is available in different languages although we have only worked with the Spanish and English sections. Table A.3

	Training		**Development**		**Test**	
	Catalan	Spanish	Catalan	Spanish	Catalan	Spanish
Number of sentences	$4.65M$		1000		2000	
Running words	$96.94M$	$96.86M$	$25.64K$	$24.45K$	$46.54K$	$46.41K$
Vocabulary	$1.28M$	$1.23M$	$6.70K$	$6.89K$	$11.22K$	$11.09K$
Avg. words per sentence	21.07	21.05	25.65	24.46	23.27	23.21

Table A.1: Statistics for the 2000-2007 "El Periódico" corpus

	Catalan	Spanish
Number of sentences	155K	
Running words	3.43M	3.43M
Vocabulary	187K	184K
Avg. words per sentence	22.86	22.12

Table A.2: Statistics for the 2008 “El Periódico” corpus

	Training		**Development**		**Test**	
	Spanish	English	Spanish	English	Spanish	English
Number of sentences	1.63M		989		1000	
Running Words	48.03M	45.07M	30.50K	29.15K	31.12K	29.56K
Vocabulary	163.66K	120.08K	5.27K	4.50K	5.19K	4.36K
Average wrd/sent	29.32	27.52	30.85	29.48	30.82	29.27

Table A.3: Statistics for Europarl.v5 corpus

shows the statistics for version 5 of the corpus while Table A.4 shows the same for version 7. The Europarl corpus is the main material used for building the baseline during Shared Tasks of the annual Workshop of Statistical Machine Translation (WMT). Apart from the parallel data used to train SMT system, Section 3.4.2.1 used a manually aligned corpus of 500 sentences from Europarl. Statistics of this manually aligned corpus can be seen in Table A.5.

A.3 News Commentary

The News Commentary corpus is also used during the annual Workshop of Statistical Machine Translation (WMT) for its Shared Tasks. It includes both parallel and monolingual data. In this thesis, we worked with the corpora from WMT 2010 and WMT 2012. Tables A.6 and A.7 shows the statistics of them for English and Spanish parallel and monolingual corpora respectively. Table A.7 also includes in its final row the statistics for the Gigaword English corpus, used to train the language model in 2012.

A.4 United Nations

Also available on WMT's Shared Task, the United Nations Corpus is a large parallel corpus that is commonly used in baseline systems for the WMT's Translation Task, together with the Europarl corpus.

	English	Spanish
Number of sentences	1.90M	
Running words	49.40M	52.66M
Vocabulary	124k	154k
Avg. words per sentence	26.05	27.28

Table A.4: Statistics for Europarl.v7 corpus

	Tuning		Testing	
	English	Spanish	English	Spanish
Number of sentences	100		400	
Running words	2.86K	3.02K	11.79K	12.49K
Vocabulary	1.04K	1.12K	2.69K	3.15K
Avg. words per sentence	28.62	30.22	29.48	31.23

Table A.5: Manually aligned Europarl corpus

WMT 2010

	Training		**Development**		**Test**	
	English	Spanish	English	Spanish	English	Spanish
Number of sentences	69.12K		1729		2525	
Running words	1.47M	1.61M	34.77K	37.09K	65.59K	70.34K
Vocabulary	37.81K	47.59K	6.19K	7.02K	8.90K	10.47K
Avg. words per sentence	21.32	23.31	20.11	21.45	25.98	27.86

WMT 2012

	Training		**Development**		**Test**	
	English	Spanish	English	Spanish	English	Spanish
Number of sentences	153K		7567		3003	
Running words	3.73M	4.33M	189.01K	202.80K	74.73K	81.01K
Vocabulary	62.70K	73.97K	18.61K	21.75K	10.82K	12.16K
Avg. words per sentence	24.20	28.09	24.98	26.80	24.88	26.98

Table A.6: Statistics for news parallel corpus

		Number of sentences	Running words	Vocabulary	Avg. words per sentence
News 2007	English	3.79M	90.25M	711.55K	23.81
	Spanish	0.05M	1.33M	64.10K	26.6
News 2008	English	13.01M	308.82M	1555.53K	23.73
	Spanish	1.71M	49.97M	377.56K	29.22
News 2009	English	14.75M	384.24M	1648.05K	26.05
	Spanish	1.07M	30.57M	287.81K	28.57
News 2010	English	6.81M	158.15M	915.14K	23.22
	Spanish	0.69M	19.58M	226.76K	28.37
News 2011	English	13.46M	312.50M	1345.79K	23.21
	Spanish	5.11M	151.06M	668.63K	29.56
Gigaword	English	22.52M	657.88M	3860.67K	29.21

Table A.7: Statistics for the monolingual news corpora

		Number of sentences	Running words	Vocabulary	Avg. words per sentence
UN 2010	English	6.22M	164.44M	1.28M	26.43
	Spanish		190.62M	1.39M	30.63
UN 2012	English	11.20M	315.90M	767.12K	28.20
	Spanish		372.21M	725.73K	33.23

Table A.8: Statistics of United Nations corpora

	Training		**Development**		**Test**	
	English	Spanish	English	Spanish	English	Spanish
Number of sentences	1.71M		993		1700	
Running words	46M	47M	24K	25K	40K	42K
Vocabulary	97K	174K	5.4K	5.9K	6.8K	7.9K
Avg. words per sentence	26.84	27.88	24.30	25.74	23.40	24.25

Table A.9: Statistics for the corpora used in the controlled feedback experiment

Table A.8 shows the main statistics of it for the years 2010 and 2012. The former was used to build the language model in Section 4.5.1 and the latter for the language model in Section 4.5.4.

A.5 Corpora used on the Feedback Experiments

The feedback experiments included a controlled enviroment, where the feedback was gathered from human experts, and a open enviroment, where the baseline system was available publicly through a website and its feedback was gathered from real users, through setences coming from any domain.

The controlled enviroment experiment (in which the post-editions were given by researchers) used a combination of Europarl+News for training baseline system, and a News Commentary subset for development and test. The statistics of the training, development and test material can be seen in Table A.9 and Table A.10 shows the statistics for the Augmented Corpus.

As for the experiment with real feedback, the data came from the Reverso.net website. Reverso[1], a private French company with SMT systems available in the internet for multiple language pairs, was a member of the European Project FAUST. In collaboration with them, we built an Augmented Corpus

[1] http://www.reverso.net

	Input (English)	Translation (Spanish)	Post-edition (Spanish)
Number of sentence	1800		
Running words	53K	55K	57K
Vocabulary	8.5K	9.2K	10.2K
Avg. words per sentence	29.24	30.41	31.15

Table A.10: Statistics for the Augmented Corpus built with the material of the Quality Estimation Task during the WMT 2012

	Input (English)	Translation (Spanish)	Post-edition (Spanish)
Number of sentence	6610		
Running words	43.31K	44.61K	47.79K
Vocabulary	8.24K	9.14K	10.42K
Avg. words per sentence	6.55	6.75	7.23

Table A.11: Statistics for the Augmented Corpus collected from the Reverso.net website

Development

FAUST	English (raw)	English (clean)	Spanish
Number of sentence	999		
Running words	10.31K	10.34K	10.67K
Vocabulary	4.24K	4.33K	4.61K
Avg. words per sentence	10.32	10.35	10.69

Test

FAUST	English (raw)	English (clean)	Spanish
Number of sentence	998		
Running words	9.95K	10.00K	10.33K
Vocabulary	4.18K	4.22K	4.49K
Avg. words per sentence	9.97	10.03	10.35

Table A.12: Statistics for the FAUST corpus provided for development and testing

consisting of input sentences coming from user's request, the automatic translation obtained with our system and the post-edited translation of those sentences, also provided by internet users. Statistics for this Augmented Corpus can be seen in Table A.11.

Apart from the 6K sentences of the Augmented Corpus, the FAUST project provided us with another corpus for development and testing. This corpus to original English requests (named English raw), gathered from internet users of Reverso; a clean version of those English sentences, with syntactic errors, typos and slangs manually fixed (named English clean) and their corresponding Spanish translation reference, provided by human translators. Table A.12 shows the statistics of this corpus.

Finally, we also worked with a monolingual Spanish corpus collected from user requests coming from different SMT systems where the source side was Spanish. This corpus was also provided by Reverso and it was used to build a language model which was later combined with others during the feedback experiment of Section 4.5.6.

	Spanish
Number of sentence	98.19K
Running words	1.00M
Vocabulary	139.16K
Avg. words per sentence	10.24

Table A.13: Spanish monolingual corpus from user requests, used to build a language model

Appendix B

Publications

2013

- Alberto Barrón-Cedeño, Lluís Màrquez, Carlos A. Henríquez Q., Lluís Formiga, Enrique Romero and Jonathan May. Identifying Useful Human Correction Feedback from an On-line Machine Translation Service. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. pp. 2057-2063. August 3-9, 2013, Beijing, China.

2012

- Lluís Formiga, Carlos A. Henríquez Q., Adolfo Hernández, José B.Mariño, Enric Monte, and José A.R. Fonollosa. The TALP-UPC Phrase-based Translation Systems for WMT12: Morphology Simplification and Domain Adaptation. In Proceedings of the Seventh Workshop on Statistical Machine Translation, June 7-8, 2012, Montréal, Canada; pp.275-282.
- Marta R. Costa-jussà, Carlos A. Henríquez Q. and Rafael E. Banchs. Evaluating Indirect Strategies for Chinese-Spanish Statistical Machine Translation. Journal of Artificial Intelligence Research. Volume 45, pages 761-780, 2012.

2011

- Mireia Farrús, Marta R. Costa-jussà, José B. Mariño, Marc Poch, Adolfo Hernández, Carlos A. Henríquez Q. and José A. R. Fonollosa. Overcoming Statistical Machine Translation Limitations: Error Analysis. Language Resources and Evaluation, 1–28, February, 2011.
- Carlos A. Henríquez Q., José B. Mariño and Rafael E. Banchs. Deriving Translation Units Using Small Additional Corpora. Proceedings of the 15th Conference of the European Association for Machine Translation, 121–128, May, 2011.

- Marta R. Costa-jussà, Carlos A. Henríquez Q. and Rafael E. Banchs. Enhancing Scarce-resource Language Translation Through Pivot Combinations. In Proceedings of the 5th International Joint Conference on Natural Language Processing. pp. 1361–1365, Chiang Mai, Thailand.

- Marta R. Costa-jussà, Carlos A. Henríquez Q., Rafael E. Banchs and José B. Mariño. Evaluating Indirect Strategies for Chinese-Spanish Statistical Machine Translation with English as Pivot Language. Procesamiento del Lenguaje Natural 47, 119–126, September, 2011.

- Carlos A. Henríquez Q., Marta R. Costa-jussà, Rafael E. Banchs, Lluis Formiga and José B. Mariño. Pivot Strategies as an Alternative for Statistical Machine Translation Tasks Involving Iberian Languages. Workshop on Iberian Cross-Language NLP tasks, September, 2011.

2010

- Carlos A. Henríquez Q., Marta R. Costa-jussà, Vidas Daudaravicius, Rafael E. Banchs and José B. Mariño. Using Collocation Segmentation to Augment the Phrase Table. Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, 104–108, July, 2010.

- Carlos A. Henríquez Q., Marta R. Costa-jussà, Vidas Daudaravicius, Rafael E. Banchs and José B. Mariño. UPC-BMIC-VDU System Description for the IWSLT 2010: Testing Several Collocation Segmentations in a Phrase-based SMT system. Proceedings of the International Workshop on Spoken Language Translation, 189–195, December, 2010.

2009

- José A. R. Fonollosa, Maxim Khalilov, Marta R. Costa-jussà, Carlos A. Henríquez Q., Adolfo Hernández and Rafael E. Banchs. The TALP-UPC Phrase-based Translation System for EACL-WMT 2009. Proceedings of the Fourth Workshop on Statistical Machine Translation, 85–89, March, 2009.

- Marc Poch, Mireia Farrús, Marta R. Costa-jussà, José B. Mariño, Adolfo Hernández, Carlos A. Henríquez Q. and José A. R. Fonollosa. The TALP On-line Spanish-Catalan Machine-translation System. Speech and Language Technologies for Iberian Languages, 105–105, September, 2009.

- Carlos A. Henríquez Q. and Adolfo Hernández. A N-gram based Statistical Machine Translation Approach for Text Normalization on Chat-speak Style Communications. Proceedings of the CAW2 Workshop, June, 2009.

2008

- Maxim Khalilov, Marta R. Costa-jussà, Carlos A. Henríquez Q., José A. R. Fonollosa, Adolfo Hernández, José B. Mariño, Rafael E. Banchs, Boxing Chen, Min Zhang, Aiti Aw and Haizhou Li. The TALP & I2R SMT Systems for IWSLT 2008. In Proceeding of the International Workshop on Spoken Language Translation. 2008, 116–123, 2008.

- Maxim Khalilov, Marta R. Costa-jussà, Josep M. Crego, Carlos A. Henríquez Q., Patrik Lambert, José A. R. Fonollosa, Rafael E. Banchs, José B. Mariño and Adolfo Hernández. The TALP-UPC *N*gram-Based Statistical Machine Translation for ACL-WMT 2008. Third ACL Workshop of Statistical Machine Translation. WMT 2008, June, 2008.

- Carlos A. Henríquez Q., Maxim Khalilov, José B. Mariño and Nerea Ezeiza. The AVIVAVOZ Phrase-based Statistical Machine Translation System for ALBAYZIN 2008. In Proceedings de las V Jornadas en Tecnología del Habla – the V Biennial Workshop on Speech Technology, 123–125, 2008.

Bibliography

[1] Adomavicius, G., and Tuzhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering 17*, 6 (2005), 734–749.

[2] Bacchiani, M., and Roark, B. Unsupervised Language Model Adaptation. In *Acoustics, Speech, and Signal Processing* (2003).

[3] Banchs, R. E., and Costa-jussà, M. R. A Semantic Feature for Statistical Machine Translation. In *Proceedings of SSST-5, Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation* (Portland, Oregon, USA, 2011), pp. 126–134.

[4] Banerjee, P., Du, J., Naskar, S. K., Li, B., Way, A., and Van Genabith, J. Combining Multi-Domain Statistical Machine Translation Models using Automatic Classifiers. In *Proceedings of The Ninth Conference of the Association for Machine Translation in the Americas* (2010).

[5] Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. Domain Adaptation in Statistical Machine Translation of User-Forum Data using Component-Level Mixture Modelling. In *Proceedings of the Machine Translation Summit XIII* (2011).

[6] Banerjee, S., and Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization* (2005).

[7] Barrón Cedeño, A., Màrques, L., Henríquez Q., C. A., Formiga, L., Romero, E., and May, J. Identifying Useful Human Correction Feedback from an On-line Machine Translation Service. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence* (2013), pp. 2057–2063.

[8] Bayes, T., and Price, R. An Essay towards solving a Problem in the Doctrine of Chance. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, M. A. and F. R. S. *Philosophical Transactions of the Royal Society of London 53* (1763), 370–418.

[9] Berger, A. L., Della Pietra, S. A., and Della Pietra, V. J. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics 22*, 1 (March 1996), 39–72.

[10] Bertoldi, N., Haddow, B., and Fouet, J.-B. Improved Minimum Error Rate Training in Moses. In *The Prague Bulletin of Mathematical Linguistics* (2009), pp. 1–11.

[11] Blain, F., Schwenk, H., and Senellar, J. Incremental adaptation using translation information and post-editing analysis. In *IWSLT-2012: 9th International Workshop on Spoken Language Translation* (Hong Kong, December 2012), pp. 229–236.

[12] Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Rossin, P. S. A Statistical Approach to Machine Translation. *Computational Linguistics 16* (1990), 79–85.

[13] Brown, P. F., Pietra, V. J., Pietra, S. A. D., and Mercer, R. L. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics 19* (1993), 263–311.

[14] Callison-Burch, C., Koehn, P., Fordyce, C. S., and Monz, C. *Proceedings of the Second Workshop on Statistical Machine Translation.* Association for Computational Linguistics, Prague, Czech Republic, June 2007.

[15] Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the 7th Workshop on Statistical Machine Translation* (Montreal, Canada, June 7–8 2012), pp. 10–51.

[16] Cherry, C., and Foster, G. Batch Tuning Strategies for Statistical Machine Translation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Montreal, Canada, June 2012), pp. 427–436.

[17] Chiang, D. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL) 2005* (2005), pp. 263–270.

[18] Chiang, D. Hierarchical Phrase-Based Translation. *Computational Linguistics 33*, 2 (2007), 201–228.

[19] Chiang, D. Hope and Fear for Discriminative Training of Statistical Translation Models. *Journal of Machine Learning Research 13* (2012), 1159–1187.

[20] Chin, J., and Rosart, D. Machine Translation Feedback. In *United States Patent* (August 2008), no. 20080195372.

[21] Clarkson, P., and Robinson, A. Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing* (1997), p. 799.

[22] Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement 20 (1)* (1960), 37–46.

[23] Crego, J. M., de Gispert, A., and Mariño, J. B. An Ngram-based Statistical Machine Translation Decoder. In *Proceedings of 9th European Conference on Speech Communication and Technology (Interspeech)* (2005).

[24] Crego, J. M., and Mariño, J. B. Syntax-enhanced N-gram-based SMT. In *MT Summit XI* (Copenhagen, Denmark, 2007).

[25] Crego, J. M., and Yvon, F. Improving Reordering with Linguistically Informed Bilingual N-grams. In *Coling 2010: 23rd International Conference on Computational Linguistics* (Beijing International Convention Center, Beijing, China, August 2010), pp. 23–27.

[26] Crego, J. M., Yvon, F., and Mariño, J. B. Ncode: an Open Source Bilingual N-gram SMT Toolkit. *Prague Bulletin of Mathematical Linguistics 96* (2011), 49–58.

[27] Cristianini, N., and Shawe-Taylor, J. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[28] Dobrinkat, M. Domain Adaptation in Statistical Machine Translation Systems via User Feedback. Master's thesis, Helsinki University of Technology, November 2008.

[29] Doddington, G. Automatic Evaluation Of Machine Translation Quality Using N-gram Cooccurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research (HLT)* (2002).

[30] Eck, M., Vogel, S., and Waibel, A. Language Model Adaptation for Statistical Machine Translation Based on Information Retrieval. In *Proceedings of Fourth International Conference on Language Resources and Evaluation (LREC)* (2004).

[31] Farrús, M., Costa-jussà, M. R., Poch, M., Hernández, A., and Mariño, J. B. Improving a Catalan-Spanish Statistical Translation System using Morphosyntactic Knowledge. In *Proceedings of European Association for Machine Translation 2009* (2009).

[32] Federico, M., Bertoldi, N., and Cettolo, M. IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models. In *Proceedings of Interspeech* (Brisbane, Australia, 2008).

[33] Formiga, L., and Fonollosa, J. A. R. Dealing with Input Noise in Statistical Machine Translation. In *Proceedings of the 24th International Conference on Computational Linguistics* (2012).

[34] Formiga, L., Hernández, A., Mariño, J. B., and Monte, E. Improving English to Spanish Out-of-Domain Translations by Morphology Generalization and Generation. In *Proceedings of the Monolingual Machine Translation-2012 Workshop Collocated with the Tenth Biennial Conference of the Association for Machine Translation in the Americas (AMTA)* (2012).

[35] Foster, G., and Kuhn, R. Mixture-Model Adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation* (2007), pp. 128–135.

[36] Fraser, A., and Marcu, D. Measuring Word Alignment Quality for Statistical Machine Translation. *Computational Linguistics 33*, 3 (August 2007), 293–303.

[37] Galley, M., and Manning, C. D. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (Honolulu, Haway, October 2008), pp. 848–856.

[38] Gao, Q., and Vogel, S. Parallel Implementations of Word Alignment Tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing* (Columbus, Ohio, June 2008), Association for Computational Linguistics, pp. 49–57.

[39] Giménez, J., and Màrquez, L. Asiya: An Open Toolkit for Automatic Machine Translation (Meta-)Evaluation. *The Prague Bulletin of Mathematical Linguistics*, 94 (2010), 77–86.

[40] Good, I. J. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika 40* (1953), 237–264.

[41] Haffari, G., Roy, M., and Sarkar, A. Active Learning for Statistical Phrase-based Machine Translation. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Morristown, NJ, USA, 2009), Association for Computational Linguistics, pp. 415–423.

[42] Hildebrand, A. S., Eck, M., Vogel, S., and Waibel, A. Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval. In *EAMT 2005 Conference Proceedings* (2005).

[43] Hopkins, M., and May, J. Tuning as Ranking. In *Conference on Empirical Methods in Natural Language Processing* (Edinburgh, Scotland, UK, July 2011), pp. 1352–1362.

[44] Huang, L., and Chiang, D. Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (Prague, Czech Republic, June 2007), pp. 144–151.

[45] Iglesias, G., de Gispert, A., Banga, E. R., and Byrne, W. Rule Filtering by Pattern for Efficient Hierarchical Translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL* (Athens, Greece, March 2009), pp. 380–388.

[46] Isabelle, P., Goutte, C., and Michel, S. Domain Adaptation of MT Systems Through Automatic Post Editing. In *Machine Translation Summit XI* (2007).

[47] Khalilov, M. *New Statistical and Syntactic Models for Machine Translation.* PhD thesis, Universitat Politécnica de Catalunya, 2009.

[48] Khalilov, M., Costa-jussà, M. R., Henríquez Q., C. A., Fonollosa, J. A., Hernández H., A., Mariño, J. B., Banchs, R. E., Chen, B., Zhang, M., Aw, A., and Li, H. The TALP & I2R SMT systems for IWSLT 2008. In *Proceedings of the International Workshop on Spoken Language Translation* (Hawai,USA, October 2008).

[49] Kneser, and Ney. Improved Backing-off for M-gram Language Modeling. In *IEEE International Conference on Acoustics, Speech and Signal Processing* (Detroit, MI, May 1995), pp. 49–52.

[50] Knight, K. A Statistical MT Tutorial Workbook. August 1999.

[51] Koehn, P. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2004), pp. 388–395.

[52] Koehn, P. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Machine Translation Summit* (2005).

[53] Koehn, P. *Statistical Machine Translation*. Cambridge University Press, 2010.

[54] Koehn, P., Amittai, A., Birch, A., Callison-Burch, C., Osborne, M., and Talbot, D. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Languages Translation* (Pittsburgh, October 2005).

[55] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions* (Morristown, NJ, USA, 2007), pp. 177–180.

[56] Koehn, P., and Schroeder, J. Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation* (2007).

[57] Kristie Seymore, R. R. Using story topics for language model adaptation. *George Kokkinakis, Nikos Fakotakis, and Evangelos Dermatas, editors, EUROSPEECH* (1997).

[58] Lewis, W. D. Haitian Creole: How to Build and Ship an MT Engine from Scratch in 4 days, 17 hours, & 30 minutes. In *Proceedings of the 14th Annual Conference of the European Association for Machine Translation* (2010).

[59] Li, Z., Callison-Burch, C., Dyer, C., Khudanpur, S., Schwartz, L., Thornton, W., Weese, J., and Zaidan, O. Joshua: An Open Source Toolkit for Parsing-Based Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation* (Athens, Greece, March 2009), Association for Computational Linguistics, pp. 135–139.

[60] Liang, P., Taskar, B., and Klein, D. Alignment by Agreement. In *Proc. of the Human Language Technology Conference of the NAACL* (New York City, USA, June 2006), pp. 104–111.

[61] Mariño, J. B., Banchs, R. E., Crego, J. M., de Gispert, A., Lambert, P., Fonollosa, J. A. R., and Costa-jussà, M. R. Ngram-based Machine Translation. *Computational Linguistics 32*, 4 (2006), 527–549.

[62] Mohit, B., and Hwa, R. Localization of Difficult-to-Translate Phrases. In *ACL 2007: Proceedings of the Second Workshop on Statistical Machine Translation* (Morristown, NJ, USA, 2007), Association for Computational Linguistics, pp. 248–255.

[63] Mohit, B., Liberato, F., and Hwa, R. Language Model Adaptation for Difficult to Translate Phrases. In *Proceedings of the 13th Annual Conference of the EAMT* (2009).

[64] Nelder, J. A., and Mead, R. A Simplex Method for Function Minimization. *The Computer Journal 7*, 4 (1965), 308–313.

[65] Niehues, J., and Waibel, A. Domain Adaptation in Statistical Machine Translation using Factored Translation Models. In *Proceedings of the European Association for Machine Translation* (2010).

[66] Och, F. Minimum error rate training in statistical machine translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics* (2003), pp. 160–167.

[67] Och, F. J., and Ney, H. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics 29* (2003), 19–51.

[68] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL* (Philadephia, USA, July 2002), pp. 311–318.

[69] Pighin, D., Màrquez, L., and May, J. An Analysis (and an Annotated Corpus) of User Responses to Machine Translation Output. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)* (Istanbul, Turkey, 2012).

[70] Razmara, M., Foster, G., Sankaran, B., and Sarkar, A. Mixing Multiple Translation Models in Statistical Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (2012), pp. 940–949.

[71] Rubino, R., Huet, S., Lefèvre, F., and Linarès, G. Statistical Post-Editing of Machine Translation for Domain Adaptation. In *Proceedings of the 16th EAMT Conference* (2012).

[72] Sennrich, R. Mixture-Modeling with Unsupervised Clusters for Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 16th EAMT Conference* (2012).

[73] Snover, M., Dorr, B., and Schwartz, R. Language and Translation Model Adaptation using Comparable Corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (2008).

[74] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas* (2006).

[75] Stolcke, A. SRILM: An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing* (Denver, CO, September 2002), pp. 901–904.

[76] Tillman, C. A Unigram Orientation Model for Statistical Machine Translation. In *HLT-NAACL 2004: Human Language Technology conference and North American Chapter of the Association for Computational Linguistics annual meeting* (2004).

[77] Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., and Sawaf, H. Accelerated DP Based Search for Statistical Translation. In *Fifth European Conference on Speech Communication and Technology* (Rhodos, Greece, September 1997), pp. 2667–2670.

[78] Ueffing, N., Haffari, G., and Sarkar, A. Semi-supervised Model Adaptation for Statistical Machine Translation. *Machine Translation 21* (2007), 77–94.

[79] Vilar, D., Leusch, G., Ney, H., and Banchs, R. E. Human Evaluation of Machine Translation Through Binary System Comparisons. In *Second Workshop on Statistical Machine Translation* (Prague, Czech Republic, June 2007), pp. 96–103.

[80] Viterbi, A. J. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory IT-13*, 2 (April 1967), 260–269.

[81] Vogel, S., Ney, H., and Tillmann, C. HMM-based Word Alignment in Statistical Translation. In *The 16th International Conference on Computational Linguistics* (Copenhagen, Denmark, August 1996), pp. 836–841.

[82] Wang, W., Macherey, K., Macherey, W., Och, F., and Xu, P. Improved Domain Adaptation for Statistical Machine Translation. In *Proceedings the Tenth Biennial Conference of the Association for Machine Translation in the Americas* (2012).

[83] Weaver, W. *Translation.* MIT Press, 1955.

[84] Wuebker, J., Mauser, A., and Ney, H. Training Phrase Translation Models with Leaving-One-Out. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (Uppsala, Sweden, July 2010), pp. 475–484.

[85] Xu, J., Deng, Y., Gao, Y., and Ney, H. Domain Dependent Statistical Machine Translation. In *Machine Translation Summit* (Copenhagen, Denmark, Sept. 2007).

[86] Xu, J., Weischedel, R., and Nguyen, C. Evaluating a Probabilistic Model for Cross-lingual Information Retrieval. In *In Proceedings of SIGIR* (2001), pp. 105–110.

[87] Zens, R., Och, F. J., Ney, H., and Vi, L. F. I. Phrase-Based Statistical Machine Translation. Springer Verlag, pp. 18–32.