Advanced techniques for domain adaptation in Statistical Machine Translation

PhD Thesis

Mara Chinea Rios

Supervised by Francisco Casacuberta
and Germán Sanchis Trilles

November, 2018

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Abstract

Machine translation is a sub-field of Natural Language Processing and Artificial Intelligence that investigates the use of computers to translate a given text from one human language to another. More specifically, Statistical Machine Translation is an approach used to build these translation systems. The quality of these systems depends mostly, on the example translations used to train or adapt the models. Corpora can come from a variety of sources, many of which are not optimal for common specific domains. Hence, the primary purpose of this thesis is to find out the right data to train or adapt models from, applied to a particular domain or task. This thesis proposes different Data Selection methods to identify task-relevant translation training data from a general data pool.

As a first step, Data Selection techniques of phrase-based Statistical Machine Translation systems are presented. Some of these techniques take advantage of continuous vector space representation of words or sentences. We applied these strategies to the task of increasing the training corpus from an available pool set. By using such approaches, experimental results prove that it is possible to achieve an increase in the translation quality and at the same time a reduction of the training corpus. A further problem regarding data selection is to find out the adequate development corpus that will be used during the tuning process in the log-linear model. Focusing in this problem, we present a Development Data Selection method paying special attention to some tests where only the source translated set was available. Results confirm the robustness across different domains. The techniques proposed are effective in a phrase-based Statistical Machine Translation. Also, experiments carried out within a real envi-

ronment are very positive and demonstrate the effectiveness of such techniques.

Neural Machine Translation paradigm, in which the Statistical Machine Translation system is based on neural networks was also studied in this thesis. In this paradigm, we investigate the application of Data Selection techniques. Two different approaches are presented in order to increase the adaptability of neural machine translation systems. On the one hand, we investigated how to increase the translation quality of the system by selecting the training corpus. The training corpus are built by concatenating the real domain training corpus and the sub-corpus obtaining; both previously obtained by a selection method. On the other hand, Data Selection was used as an efficient strategy to create synthetic corpus. This synthetic corpus was employed to adapt translation models present, at the moment, in state-of-the-art neural machine translation systems. Adaptation experiments were performed in different domains and the translation results obtained are compelling for both tasks.

In addition, special attention is devoted to optimise Statistical Machine Translation log-linear weights. With this purpose, an optimisation method that intends to increase the translation quality for a specific domain was studied. We performed experiments across different domains and compared our results with other state-of-the-art methods. Although the research performed within this topic was initially conceived as a chapter, it was finally moved to a appendix to allow the rest of the thesis to focus more intensely on its main topic: data selection.

Finally, it should be noted that the techniques developed and presented in this thesis may be readily implemented within a real translation scenario, in which a statistical machine translation system is used to solve a real problem.

Keywords: Statistical Machine Translation, phrase-based, Neural Machine Translation, domain adaptation, Data Selection, continuous vector space representation, word embeddings.

Resumen

La Traducción Automática Estadística es un sup-campo de la lingüística computacional que investiga como emplear los ordenadores en el proceso de traducción de un texto de un lenguaje humano a otro. La traducción automática estadística es el enfoque más popular que se emplea para construir estos sistemas de traducción automáticos. La calidad de dichos sistemas depende en gran medida de los ejemplos de traducción que se emplean durante los procesos de entrenamiento y adaptación de los modelos. Los conjuntos de datos empleados son obtenidos a partir de una gran variedad de fuentes y en muchos casos puede que no tengamos a mano los datos más adecuados para un dominio específico. Dado este problema de carencia de datos, la idea principal para solucionarlo es encontrar aquellos conjuntos de datos más adecuados para entrenar o adaptar un sistema de traducción para un dominio o tarea específico. En este sentido, esta tesis propone un conjunto de técnicas de selección de datos que identifican los datos bilingües más relevantes para una tarea extraídos de un gran conjunto de datos. Algunas de estas técnicas aprovechan las ventajas que presenta la representación vectorial del texto en un espacio continuo.

Como primer paso en esta tesis, las técnicas de selección de datos son aplicadas para mejorar la calidad de la traducción de los sistemas de traducción automática estadísticos bajo el paradigma basado en frases. Estas técnicas se basan en el concepto de representación continua de las palabras o las oraciones en un espacio vectorial. Las técnicas desarrolladas fueron aplicadas a la tarea de aumentar el tamaño de un conjunto de entrenamiento pequeño que pertenece al dominio de la tarea. Los resultados experimentales presentados para esta tarea

demuestran que es posible lograr un aumento de la calidad de la traducción y al mismo tiempo una reducción significativa en el tamaño del conjunto de entrenamiento. Otra tarea dentro de este paradigma fue seleccionar los mejores conjuntos de desarrollo que se emplean durante el proceso de ajuste de pesos del modelo log-lineal. Enfocándonos en este problema, en esta trabajo se presentan diferentes métodos para la selección de los conjuntos de desarrollo, prestando especial atención a aquellos casos que sólo tenemos disponible el conjunto de oraciones a traducir. Los resultados experimentales demuestran que las técnicas utilizadas son efectivas para diferentes lenguajes y dominios. Además, los experimentos llevados a cabo en un entorno real son muy positivos y demuestran la efectividad de los métodos.

El paradigma de Traducción Automática Neuronal también fue aplicado en esta tesis. Dentro de este paradigma, investigamos la aplicación que pueden tener las técnicas de selección de datos anteriormente validadas en el paradigma basado en frases. El trabajo realizado se centró en la utilización de dos tareas diferentes de adaptación del sistema. Por un lado, investigamos cómo aumentar la calidad de traducción del sistema, aumentando el tamaño del conjunto de entrenamiento. Los conjuntos de entrenamiento que se emplearon se construyeron concatenando el conjunto de entrenamiento de dominio y el sub-conjunto obtenido por un método de selección. Por otro lado, el método de selección de datos se empleó como una estrategia eficiente para crear un conjunto de datos sintéticos. Estos conjuntos sintéticos fueron empleados para adaptar un sistema de traducción automática neuronal general al dominio que deseábamos. Los experimentos se realizaron para diferentes dominios y los resultados de traducción obtenidos son convincentes para ambas tareas.

Además de la tarea de selección de datos, se prestó atención al proceso de optimización de los pesos del modelo log-lineal. Con este propósito, se estudió un método de optimización que pretende aumentar la calidad de la traducción de un sistema para un dominio específico. Los experimentos fueron realizados para diferentes dominios y comparamos los resultados obtenidos con nuestro método con los resultados obtenidos por otros métodos de vanguardia. Aunque la investigación realizada dentro de este tema se concibió inicialmente como un capítulo de esta tesis, finalmente se trasladó a un apéndice

para permitir que el resto de la tesis se centre más intensamente en su tema principal: selección de datos.

Finalmente, cabe señalar que las técnicas desarrolladas y presentadas a lo largo de esta tesis pueden implementarse fácilmente dentro de un escenario de traducción real; donde el sistema de traducción esta diseñado para resolver un problema real existente.

Palabras clave: Traducción Automática Estadística, modelos basados en frases, Traducción Automática Neuronal, adaptación de dominios, Selección de Datos, representación vectorial en espacio continuo.

Resum

La Traducció Automàtica Estadística és un subcamp la lingüística computacional que investiga com emprar els ordinadors en el procés de traducció d'un text d'un llenguatge humà a un altre. La traducció automática estadística és l'enfocament més popular que s'empra per a construir aquests sistemes de traducció automàtics. La qualitat de aquests sistemes depén en gran mesura dels exemples de traducció que s'empren durant els processos d'entrenament i adaptació dels models. Els conjunts de dades emprades són obtinguts a partir d'una gran varietat de fonts i en molts casos pot ser que no tinguem a mà les dades més adequades per a un domini específic. Donat aquest problema de manca de dades, la idea principal per a solucionar-ho és trobar aquells conjunts de dades més adequades per a entrenar o adaptar un sistema de traducció per a un domini o tasca específic. En aquest sentit, aquesta tesi proposa un conjunt de tècniques de selecció de dades que identifiquen les dades bilingües més rellevants per a una tasca extreta d'un gran conjunt de dades. Algunes d'aquestes tècniques aprofiten els avantatges que presenta la representació vectorial del text en un espai continu.

Com a primer pas en aquesta tesi, les tècniques de selecció de dades són aplicades per a millorar la qualitat de la traducció dels sistemes de traducció automàtica estadístics sota el paradigma basat en frases. Aquestes tècniques es basen en el concepte de representació contínua de les paraules o les oracions en un espai vectorial. Les tècniques desenvolupades van ser aplicades a la tasca d'augmentar la grandària d'un conjunt d'entrenament xicotet que pertany al domini de la tasca. Els resultats experimentals presentats per a aquesta tasca demostren que és possible aconseguir un augment de la quali-

tat de la traducció i al mateix temps una reducció significativa en la grandària del conjunt d'entrenament. Una altra tasca dins d'aquest paradigma va ser seleccionar els millors conjunts de desenvolupament que s'empren durant el procés d'ajust de pesos del model log-lineal. Enfocant-nos en aquest problema, en aquest treball es presenten diferents mètodes per a la selecció dels conjunts de desenvolupament, prestant especial atenció a aquells casos que només tenim disponible el conjunt d'oracions a traduir. Els resultats experimentals demostren que les tècniques utilitzades són efectives per a diferents llenguatges i dominis. A més, els experiments duts a terme en un entorn real són molt positius i demostren l'efectivitat dels mètodes.

El paradigma de Traducció Automàtica Neuronal també va ser aplicat en aquesta tesi. Dins d'aquest paradigma, investiguem l'ús que poden tenir les tècniques de selecció de dades anteriorment validades en el paradigma basat en frases. El treball realitzat es va centrar en la utilització de dues tasques diferents d'adaptació del sistema. D'una banda, investiguem com augmentar la qualitat de traducció del sistema, augmentant la grandària del conjunt d'entrenament. Els conjunts d'entrenament que es van emprar es van construir concatenant el conjunt d'entrenament de domini i el subconjunt obtingut per un mètode de selecció. D'altra banda, el mètode de selecció de dades es va emprar com una estratègia eficient per a crear un conjunt de dades sintètiques. Aquests conjunts sintètics van ser emprats per a adaptar un sistema de traducció automàtica neuronal general al domini que desitjàvem. Els experiments es van realitzar per a diferents dominis i els resultats de traducció obtinguts són convincents per a ambdues tasques.

A més de la tasca de selecció de dades, es va parar esment al procés d'optimització dels pesos del model log-lineal. Amb aquest propòsit, es va estudiar un mètode d'optimització que pretén augmentar la qualitat de la traducció d'un sistema per a un domini específic. Els experiments van ser realitzats per a diferents dominis i comparem els resultats obtinguts amb el nostre mètode amb els resultats obtinguts per altres mètodes d'avantguarda. Encara que la investigació realitzada dins d'aquest tema es va concebre inicialment com un capítol d'aquesta tesi, finalment es va traslladar a un apèndix

per a permetre que la resta de la tesi se centre més intensament en el seu tema principal: selecció de dades.

Finalment, cal assenyalar que les tècniques desenvolupades i presentades al llarg d'aquesta tesi poden implementar-se fàcilment dins d'un escenari de traducció real; on el sistema de traducció aquesta dissenyat per a resoldre un problema real existent.

Paraules clau: Traducció Automàtica Estadística, models basats en frases, Traducció Automàtica Neuronal, adaptatió de dominis, Selecció de Dades, representació vectorial en espai continu.

Acknowledgments

Ahora que me encuentro al final de este largo viaje y que empiezo a ver la luz al final del tunel me gustaría agradecer el apoyo de todos aquellos, que de una forma u otra, han ayudado a lo largo de este viaje.

En primer lugar, me gustaría dar las gracias a mis directores. A **Francisco Casacuberta**, que me dió la oportunidad de dedicarme a la traducción automática y realizar esta tesis, y a **Germán Sanchis Trilles**, que sin su guía y entusiasmo esta tesis no hubiera sido posible. Me gustaría agradecer también a todos mis compañeros del PRHLT por el extraodinario ambiente de trabajo y todos los momentos de risas durante todos estos años. En particular, a mis compañeros del grupo de traducción por la ayuda recibida y por los artículos en los que hemos colaborado.

Fuera del mundo de la investigación, me gustaría agradecer a varias personas. Tal vez la persona que más me ha apoyado durante el desarrollo de la tesis es **David**, gracias por tu paciencia y cariño. A mi **hermana Rocío** le quiero dedicar un agradecimiento especial, por su incondicional ayuda que queda reflejada en cada capítulo de esta tesis. Por último quisiera agradecer, **a mis padres**, por todo lo que me han enseñado, por su apoyo en todo momento a pesar de la distancia.

Lo único que puedo decir a todos es ¡MUCHAS GRACIAS!

Valencia, Julio del 2018

Preface

The phenomenon of globalisation, together with communication age, have increased the interest in different research fields, in particular, in Machine Translation (MT). Nevertheless, initial efforts to build automated translation engines have started almost as soon as computers came into existence. The idea behind MT is to make use of computers to translate a given text or speech from one language to another. Many different paradigms have been developed across the years. In general terms, Statistical Machine Translation (SMT) is an approach to machine translation that is characterized by the use of machine learning methods. Statistical Machine Translation has come to dominate academic machine translation research and also has gained a share of the commercial market.

The performance of any SMT system depends largely on the quality and quantity of the bilingual corpus on which the system is built. If we were able to train a general system on a sufficiently general corpus it would be possible, for instance, to translate a newspaper editorial, updates on a social network, tweets, or any other text available. However, machine translation is also needed in some fields where the amount of data available is less abundant, or even does not exist yet. This can be the case of specific domains or languages such as medicine or in the technical domain, where translation quality is still critical. We can use a general translation system to translate specific documents, but translation quality will most likely suffer when context is a problem.

Hence, the main objective of this thesis is to adapt SMT systems to a specific domain, improving translation quality with a strong empha-

sis on those domains with low resources. More precisely, the scientific contributions of this thesis can be divided in three groups as follows:

1. **Data selection in phrase-based SMT (PBSMT) systems:** Novel data selection techniques are presented for this MT paradigm. These methods are employed in different tasks, e.g., to increase the size of the training corpus or select the development corpus. Experimental results are reported on several domains and language pairs.

2. **Data selection in Neural Machine Translation (NMT) systems:** Given the good results obtained with data selection methods in phrase-based SMT, we also study the use of these techniques in NMT. First, DS techniques are used to increase the size of the training corpus, employed during the training process of a NMT system. Then, we present a novel data selection technique to create synthetic corpora for the purpose of adapting an NMT system. Experiments are conducted in an NMT setting, involving several different language pairs and corpora. We also compare the results achieved with those obtained with phrase-based SMT.

Finally, we introduce an appendix titled "Log-linear weight optimization". In this appendix, a novel log-linear weight optimization method is presented. Such technique relies on the concept of discriminative ridge regression for obtaining the best weight vector required by the log-linear model in phrase-based SMT. Exhaustive experimental results are reported on several domains and language pairs. Although the research performed within this appendix was initially conceived as a chapter, it was finally moved to the appendix to allow the rest of the thesis to focus more intensely on its main topic: data selection.

Thesis Structure

This thesis has been set up in a modularised manner in order to help the reader gain a wider insight into adaptation methods for MT. The thesis has been structured as a tree, Figure 1, where each node is a

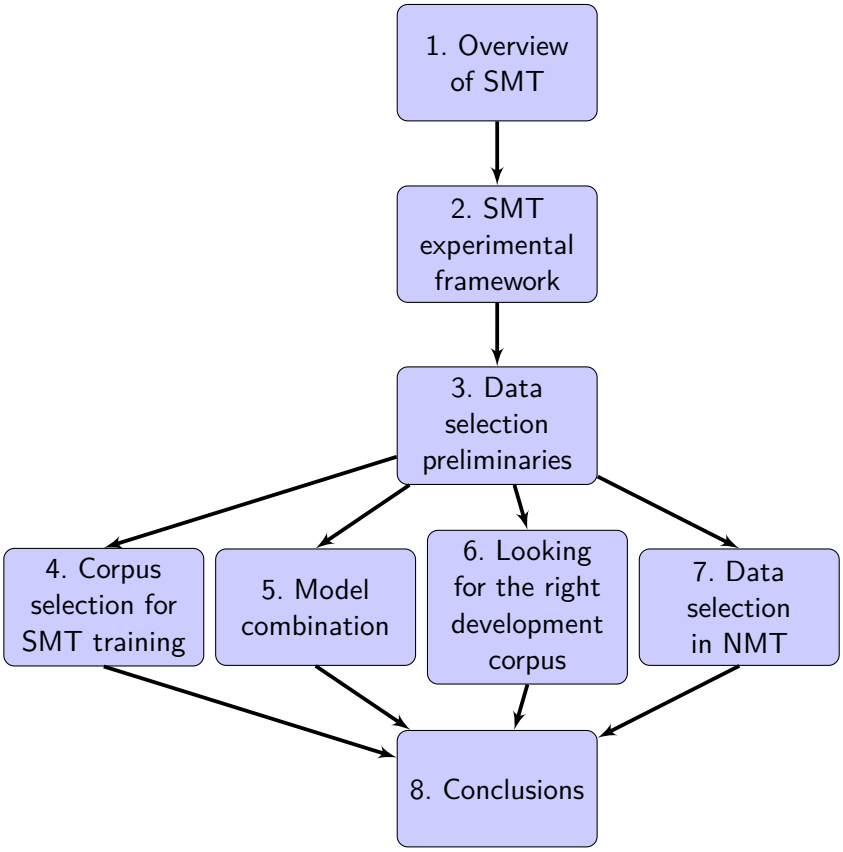


Figure 1. Diagram to describe the chapters thesis structures.

different chapter. The above contributions are organised in 8 chapters and 1 appendix that cover most of the work developed in this thesis. A sequential reading of the document is recommended if the reader wishes to learn about the complete work. However, in case the reader is only interested in a specific research topic, she/he can also read only the chapters that are related to the topic in question. A very brief summary of each chapter is given bellow:

1. **Overview of SMT:** This chapter introduces the Statistical Machine Translation paradigm, its problem statement and other important issues.

2. **SMT experimental framework:** This chapter presents the set-up employed in this thesis, description of the employed corpora, principal toolkits and automatic metrics.

3. **Data selection preliminaries:** This chapter builds up the plot by shedding some light on the background that is needed to understand the following chapters.

4. **Corpus selection for SMT training:** The data selection techniques are proposed in this chapter. In addition, experimental results are presented, across different domains and language pairs, evaluating the translation quality by means of different automatic metrics.

5. **Model combination:** Two different approaches for leveraging the selected corpora are presented. Experimental results allow to estimate the quality of the translations produced by the combined systems.

6. **Looking for the right development corpus:** The choice of development corpus has a big impact on translation quality. Hence, in this chapter different methods to find out the best development corpus are proposed.

7. **Data selection in NMT:** The data selection techniques applied in the NMT paradigm are presented in this chapter.

8. **Conclusions:** This is the final chapter, which summarises the conclusions that can be drawn from all the work described here, together with the work that still lies ahead. In addition, the most important scientific publications that have been derived from this thesis are listed.

Related Publications

All the work presented in this thesis were accepted for publication in international conferences or peer-reviewed journals:

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Bilingual Sentence Selection Strategies: Comparative and Combina-

tion in Statistical Machine Translation. In Proceedings of the IberSPEECH, pages 227–236, 2014. (Relative to Chapter 3)

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. An Empirical Analysis of Data Selection Techniques in Statistical Machine Translation. Revista Procesamiento del Lenguaje Natural, volumen 55, 2015. (Relative to Chapter 3)

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Bilingual Data Selection Using a Continuous Vector-Space Representation. In Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition, pages 95–106, 2016. (Relative to Chapters 3-6)

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Making Better Use of Data Selection Methods. In Proceeding In Proceedings of the IberSPEECH, 2016. (Relative to Chapter 4)

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Log-Linear Weight Optimization Using Discriminative Ridge Regression Method in Statistical Machine Translation. In Proceeding of the Iberian Conference on Pattern Recognition and Image Analysis, pages 32–41, 2017. (Relative to (Relative to Appendix Log-linear weight optimization)

- Á. Peris, **M. Chinea-Rios**, and F. Casacuberta. Neural Networks Classifier for Data Selection in Statistical Machine Translation. Journal The Prague Bulletin of Mathematical Linguistics, volume 108, pages 283–294, 2017. (Relative to Chapters 3-6)

- **M. Chinea-Rios**, Á. Peris, and F. Casacuberta. Adapting Neural Machine Translation with Parallel Synthetic Data. In Proceedings of the Second Conference on Machine Translation, pages 138–147, 2017. (Relative to Chapter 6)

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Discriminative ridge regression algorithm for adaptation in statistical machine translation. Journal Pattern Analysis and Applications, 2018. (Relative to Appendix Log-linear weight optimization)

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Creating the Best Development Corpus for Statistical Machine Translation Systems, In Proceeding of the Annual Conference of the European Association for Machine Translation, pages 99–108, 2018. (Relative to Chapters 5)

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Are Automatic Metrics Robust and Reliable in Specific Machine Translation Tasks? In Proceeding of the Annual Conference of the European Association for Machine Translation, pages 89–98, 2018. (Relative to Chapters 6)

Finally, there is one further publication which has been submitted to an international journal, but which has not yet been accepted:

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Vector Sentence Representation for Data Selection in Statistical Machine Translation . Journal Computer Speech and Language (submitted for revision). (Relative to Chapter 3-6)

In addition, further work carried out during the same period of time than the present thesis, but that is not directly related to the topics presented here, was published in several international conferences:

- **M. Chinea-Rios**, G. Sanchis-Trilles, D. Ortiz-Martínez, and F. Casacuberta. Online optimisation of log-linear weights in interactive machine translation. In Proceedings of the International Conference on Language Resources and Evaluation, pages 3556–3559, 2014.

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Sentence clustering using continuous vector space representation. In Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, pages 432–440, 2015.

- **M. Chinea-Rios**, G. Sanchis-Trilles, and F. Casacuberta. Domain Adaptation Problem in Statistical Machine Translation Systems. In Proceedings of the International Conference of the Catalan Association of Artificial Intelligence, pages 205–213, 2016.
- M. Domingo, **M. Chinea-Rios**, and F. Casacuberta. Historical Documents Modernization. Journal The Prague Bulletin of Mathematical Linguistics, volume 108, pages 295–306, 2017.

Contents

1 Overview of Statistical Machine Translation

* * *

"Voici mon secret. Il est très simple: on ne voit bien qu'avec le cœur. L'essentiel est invisible pour les yeux."

—ANTOINE DE SAINT-EXUPÉRY
LE PETIT PRINCE

"Here is my secret. It is very simple: you can only see with the heart. What is essential is invisible to the eye."

—GOOGLE TRANSLATOR
THE LITTLE PRINCE

* * *

1.1 Introduction

The translation of foreign language texts by computers is a field of research with more than sixty years of activity and the need to translate more documents is growing exponentially. Machine translation (MT) becomes an important area of artificial intelligence and natural language processing again after having been in a pause during years.

However, MT is an open problem for various reasons: the way to measure translation quality, the limited resources in many languages and the context problem etc..

This chapter provides, in the first place, an introduction to the general strategies that have been proposed to deal with the MT problem. We then make a summary of both the historical and current research landscape in the Statistical Machine Translation (SMT) paradigm.

Table 1.1 shows the abbreviations introduced in the current chapter, in order to facilitate a better comprehension of the text.

Table 1.1. Abbreviations used in Chapter 1.

Abbreviation	Description
NLP	Natural Language Processing
MT	Machine Translation
RBMT	Rule-Based Machine Translation
SMT	Statistical Machine Translation
PB	Phrase-based
PBSMT	Phrase-based Statistical Machine Translation
NMT	Neural Machine Translation
CVR	Continuous vector-space representation
Skip-Gram	Continuous Skip-Gram model

1.2 Machine Translation

Machine translation is a specific sub-field of Natural Language Processing (NLP), and studies the way in which automatic systems are able to automatize the translation process. MT systems translate a certain input text from one language into another, while trying to ensure that the output sentence is well structured in the target language.

Different approaches have been developed and used during recent years within different paradigms and level of success. The first and most intuitive approach to MT is the *word-for-word* translation. In simple words, it can be described as the rendering of text from the source language to the target language, one word at a time, while conveying the sense of the original sentence. While word-for-word translation is easy to implement, the order and context of the words are not included in the process. Consequently, sentences translated word-for-word are in most cases difficult to understand. The *interlingua* approach is another MT paradigm. In this approach, the text to be translated is transformed into an abstract language representation. This abstract language is called interlingua and is independent of both source and target languages. The target language is then generated from the interlingua. The principal disadvantage is to define

and create an adequate interlingua because it is difficult to apply to a wider domain.

The Rule-Based Machine Translation (RBMT) paradigm [1] is based on linguistic information about source and target languages. RBMT depends on translation rules created by human translators to generate their hypothesis. The process of creating translation rules is very costly and requires the knowledge provided by expert linguists. For this reason, RBMT systems are loosing weight in the state of the art in comparison to more robust and cost-effective approaches, such as statistical machine translation. Nevertheless, they are still in use in some commercial systems, such as Apertium [2,3] and GramTrans [4].

Corpus-based systems make use of the so-called empirical proposals of MT. The main feature of corpus-based systems is the use of sets of translation examples (also called corpus or parallel sets) from one language to another [5]. In contrast to RBMT systems which are specific for a given language pair or domain, corpus-based systems can be quickly adapted for their use on different domains or language pairs. There are different types of corpus-based systems, and can be classified in two groups: example-based machine translation systems and statistical machine translation systems.

- Example based approach: This approach to machine translation uses a set of translation examples as its main knowledge base. One important translation technology derived from the example-based approach of MT is the so-called memory-based machine translation. Memory-based translation systems allow to assist human translators in the translation process and stores user validated translations (translation memories) for its reuse in the translation of similar texts.

- Statistical machine translation: The other group of corpus-based systems, and maybe the most well-known, are statistical machine translation systems. The statistical machine translation approach [5] requires the availability of a parallel corpus containing relevant information for the translation process. In this paradigm, a corpus is used to estimate the parameters of a set of statistical models involved in the translation process. These

estimated statistical models are used to obtain the translation. The MT paradigm based on neural networks can be also classified inside this group. Due to the considerable increase in the linguistic resources, better and more complex statistical models have been obtained. They will be explained in more detail in Section 1.3.

1.3 Statistical Machine Translation

SMT systems are an important alternative concerning other machine translation paradigms as RBMT systems. The main benefit of SMT systems is that they are mathematically well founded, not language-dependent, efficient, and allows for a fast development of MT systems if sufficient parallel corpora are available. Consequently, different alternatives to RBMT are widely-studied nowadays, e.g., the neural machine translation paradigm. The SMT paradigm to MT formalises the problem of generating translations under a statistical point of view. Bilingual corpora are necessary to estimate the parameters of the statistical models involved in the translation process. Within the SMT paradigm, translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora. These models can be described as a mathematical formulation about how a sentence $\mathbf{x}$ in the source language is translated into an equivalent sentence $\mathbf{y}$ in the target language. Formally, given a source sentence $\mathbf{x} = x_1, \ldots, x_j, \ldots, x_J$ from the source language, we need to find the equivalent target sentence $\mathbf{y} = y_1, \ldots, y_i, \ldots, y_I$ from the target language, where $x_j \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ denote the source and target words, coming from the source and target vocabularies, $\mathcal{X}$ and $\mathcal{Y}$ respectively. $J = |\mathbf{x}|$ and $I = |\mathbf{y}|$ are the lengths of the source and target sentences.

From the set of all possible translation sentences in the target language, the SMT process aims to find out the sentence with the highest probability $\hat{\mathbf{y}}$ according to the following equation [6]:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x}) \tag{1.1}$$

However, $p(\mathbf{x}|\mathbf{y})$ is not simpler to estimate. Bayes [7] comes into play, because it permits to leverage a language model, too:

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, p(\mathbf{y}) \cdot p(\mathbf{x}|\mathbf{y}) \tag{1.2}$$

Equation 1.2 is known as the source-channel model for SMT [6], and it is also referred to as the *fundamental equation* of SMT. In Equation 1.2, the term $p(\mathbf{y}|\mathbf{x})$ has been decomposed in two different probabilities: the language model of the target language, $p(\mathbf{y})$, and the translation model $p(\mathbf{x}|\mathbf{y})$. The language model $p(\mathbf{y})$ measures the well-formedness of the target language sentences. The translation model $p(\mathbf{x}|\mathbf{y})$, on the other hand captures the correlation between the source and target sentences.

Word alignment models were introduced by [6]. In the inverse version of the word alignment model, a source word x_j is aligned to a set of target word positions $\mathbf{a}_j = i_1, \ldots, i_l$, following a generative perspective. Such an alignment implies that source word x_j generates target words $y_{i_1}, \ldots, y_{i_l}$. Modelling the translation process in such a way requires the use of a hidden variable $\mathbf{a}$ and $p(\mathbf{x}, \mathbf{a}|\mathbf{y})$ is the probability of translating x^J by y^I given the alignment $\mathbf{a}$, since alignments cannot be observed in the training process:

$$p(\mathbf{x}|\mathbf{y}) = \sum_{a \in \mathcal{A}(\mathbf{x},\mathbf{y})} p(\mathbf{x}, a|\mathbf{y}) \tag{1.3}$$

There are five different types of word-alignment models, International Business Machines (IBM) models, ranging from IBM 1 to the IBM 5 model with an increasing degree of complexity. In addition, other works proposed further models [8], which have also gained popularity.

An alternative to the source-channel approach is to directly model the posterior is $p(\mathbf{y}|\mathbf{x})$ in Equation 1.2 using the maximum entropy model [9–11]. This approach is usually called *log-linear model*. Log-linear models are characterized by an ensemble of feature functions $h_m(\mathbf{x}, \mathbf{y})$, where $m \in 1, \ldots, M$ is the number of features in the models. A weight λ_m is assigned to each feature representing how important the feature h_m is for the translation of $\mathbf{x}$ into $\mathbf{y}$. Formally, the transla-

tion probability is modeled as follows:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \frac{exp\ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{y}'} exp\ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{x},\mathbf{y}')} \\ &= \frac{exp\ \boldsymbol{\lambda}\cdot\mathbf{h}(\mathbf{x},\mathbf{y})}{\sum_{\mathbf{y}'} exp\ \boldsymbol{\lambda}\cdot\mathbf{h}(\mathbf{x},\mathbf{y}')} \end{aligned} \tag{1.4}$$

where $h_m(\mathbf{x},\mathbf{y})$ is a score representing an imported feature for the translation of $\mathbf{x}$ into $\mathbf{y}$ and $\boldsymbol{\lambda} = [\lambda_1,\ldots,\lambda_M]$. Similarly, as done in the source-channel approach in the Equation 1.2, the denominator can be neglected during the search process because it does not depend on translation hypothesis $\hat{\mathbf{y}}$. As a result of the previous consideration, given the Equation 1.4, the final decision rule is stated as follows:

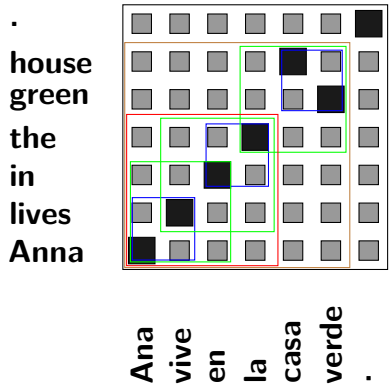
$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\ \boldsymbol{\lambda}\cdot\mathbf{h}(\mathbf{x},\mathbf{y}) \tag{1.5}$$

The use of log-linear models implied an important break-through in SMT, allowing a significant increase in the quality of the translations produced.

1.3.1 Phrase-based Statistical Machine Translation

Phrase-based (PB) models constitute the most popular instantiation of the log-linear model in SMT and constitute an alternative to overcome the limitations that the word based models [5,12–14] exhibit. PB models are based in the concept of segmenting the sentence pairs into phrases (i.e., word sequences) where the number of source phrases is equal to the number of target phrases (K) and one source segment is aligned only with one target phrase and vice versa.
The translation of the source sentence $\mathbf{x}$ into the equivalent target sentence $\mathbf{y}$ using PB models can be explained following these steps:

1. Divide the source sentence $\mathbf{x}$ into K source phrases, $\tilde{x}_1 \ldots \tilde{x}_k \ldots \tilde{x}_K$.
2. $\tilde{y}_1 \ldots \tilde{y}_k \ldots \tilde{y}_K$, translate each of the source phrases into the target phrases.
3. The target phrase translations are reordered to compose the target sentence $\hat{\mathbf{y}}$.



(a) Word matrix

Source phrase	Target phrase
Ana	Anna
vive	lives
en	in
la	the
casa	house
verde	green
.	.
Ana vive	Anna lives
vive en	live in
en la	in the
la casa verde	the green house

(b) Phrases extracted

Figure 1.1. Example of how consistent phrases are extracted from a word alignment matrix within a phrase based model.

Another important step when learning a PB model is to obtain a phrase-table. The phrase-table is a translation table containing all of the phrase pairs $\tilde{\mathbf{x}}, \tilde{\mathbf{y}}$ observed during training, and includes the values of each one of the feature functions assigned to that phrase pair.

A wide variety of heuristic techniques to produce PB model have been investigated and implemented [12]. The most commonly used PB model estimation technique [14, 15] is based on the relative frequencies of the phrase pairs that are extracted from word alignment matrices [16]. Similarly to word alignment models, PB models assume that the relationship between the source and the target phrases are explained through an alignment variable. This alignment variable summarises some of the decisions made during the generative process. Figure 1.1 shows examples of the word aligned sentence pair and the bilingual phrases extracted from this sentence. In Figure 1.1a, the alignment matrix is shown. Black squares represent word alignments, whereas extracted phrases are highlighted with a rectangle comprising one or more squares. Figure 1.1b lists the phrases that would be extracted from the matrix.

The different features $h_m(\cdot,\cdot)$ that are included into the phrase-table are:

1. Inverse translation probability, obtained by the formula

$$p(\tilde{\mathbf{x}}|\tilde{\mathbf{y}}) = \frac{Count(\tilde{x},\tilde{y})}{Count(\tilde{x})} \tag{1.6}$$

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

2. Direct translation probability, which is similar to the inverse translation probability computed in the reverse translation direction.

3. Direct and inverse lexical translation probabilities. These features were defined by [14], and attempt to account for the lexical soundness of each phrase pair.

4. The phrase penalty, which is like the word penalty feature, implementing a constant cost during decoding. The phrase penalty is accumulated per phrase.

All these features are defined over phrases, and not sentences, because they are used to construct the translation hypotheses. Hence, a sentence feature value is the product of the phrase score and the best-scoring partition of the segmentation of sentences into phrases [5].

1.3.1.1 Tuning phrase-based models

In the previous section, we discussed how bilingual phrases are extracted. At this point, it is still necessary to obtain an appropriate value for the scaling factors, or log-linear weights ($\boldsymbol{\lambda}$) present in log-linear SMT framework (see Equation 1.4). The weights $\boldsymbol{\lambda}$ adjust the importance of each single model within the specific task. This process is often called *tuning*. The main idea behind it is that good values for a certain task might not be the appropriate values for other tasks. To exemplify this, consider for instance that the original translation model has been trained on a domain in which sentences tend to be long, e.g.

in the European parliamentary debate. Then, if we intend to translate another domain in which sentences are rather short (sentences of technical manuals) we will have to adjust the weights conveniently to reflect this fact.

To this end, numerous methods have been proposed to optimize the log-linear model weights. The most popular algorithm for optimising the scaling factors $\boldsymbol{\lambda}$ is the one proposed by Koehn and Och [5, 17] commonly known as Minimum Error Rate Training (MERT) . MERT has two critical drawbacks. On the one hand, it heavily relies on having a fair amount of data available as development set. On the other hand, it only relies on the data from the development set. These two problems can produce over-fitting to the specific characteristics of the development set and such algorithms fail to provide appropriate estimates ([18–20])

Various alternatives to MERT have been proposed, motivated primarily by previous problems. For instance, [21–23] proposed the use of the Margin Infused Relaxed Algorithm (MIRA) for the task of optimising $\boldsymbol{\lambda}$. In, [24] proposed to view the tuning problem as a set of operations over a specific semiring. Alternatively, [25] proposed to address the problem as a ranking problem, where each step of the tuning procedure consists in deciding whether a given translation hypothesis should be ranked lower or higher within the set of possible hypotheses that are provided by the search procedure.

Minimum Error Rate Training algorithm MERT is initialised as follows: $\mathbf{x}_1, \ldots, \mathbf{x}_A$ denote the A input sentences of the tuning set (or development corpus). Initially, an initial weight-vector $\boldsymbol{\lambda}$ is chosen and n-best list from the decoder is obtained. Then, the iterative part of the algorithm starts. In the first run, the starting point is the initial weight-vector $\boldsymbol{\lambda}$ and in the next iterations this will be the best weight-vector from the previous iteration. After each iteration, the decoder is run again to obtain new n-best lists that are merged with the existing ones. Besides this single starting point, MERT typically uses a number of additional random points in vector space to avoid poor local optima. The iterations stop if there are no changes in the weight-vector, or if there are no new translations in the n-best list. In MERT, the goal is to minimize the error count $E(\mathbf{r}, \mathbf{y})$ by scoring translation

hypotheses against a set of reference translations $r_1, \ldots, r_A$. Assuming as in [17] that error count is additively decomposable by sentence i.e., $E(\mathbf{r}_1^A, \mathbf{y}_1^A) = \sum_a E(\mathbf{r}_a, \mathbf{y}_a)$, this results in the following optimization problem:

$$\hat{\boldsymbol{\lambda}} = \underset{\lambda}{\operatorname{argmin}} \left\{ \sum_a E(\mathbf{r}_a, \hat{\mathbf{y}}(\mathbf{x}_a; \boldsymbol{\lambda})) \right\} \tag{1.7}$$

The quality of the results obtained by MERT depends on how accurately the n-best list represents the search space of the system.

Margin Infused Relaxed Algorithm is an online version of the large margin training algorithm for structured classification [26]. It updates the log-linear weight vector $\boldsymbol{\lambda}$ according to certain margin constraints and a loss function. It is attractive to use MIRA because its weight-vector updates are in proportion to the loss incurred by misclassifying a pair of candidate translations. This method adapts a weight-vector based on how far off it is from a pair of translations and how much this will cost.

1.3.1.2 Decoding process

Once the PB models assign a score to every possible translation of a source input sentence $\mathbf{x}$, an algorithm is need for selecting and establishing which is the best candidate hypothesis $\mathbf{y}^*$. Thus, the goal of the decoding process is to find out the translation with the best score [5]. In general, decoding is a hard problem, since there is an exponential number of possible translations given a specific input sentence $\mathbf{x}$. In other words, exhaustively searching all possible translations, scoring and selecting the best translation is computationally very expensive, and decoding is actually a NP-hardproblem [27]. To overcome this issue, different heuristic search methods have been used. These heuristic methods do not guarantee that they will find the best translation $\hat{\mathbf{y}}^*$, but we hope to obtain a translation that is very close to $\mathbf{y}^*$. Typical examples for these methods are the multi-stack depth-first decoding algorithm [28] proposed by [9] for word-based models, greedy strategies [29,30] and finally, the search algorithm by [31,32], which is an adaptation of the classic algorithm for speech recognition in SMT proposed in [33].

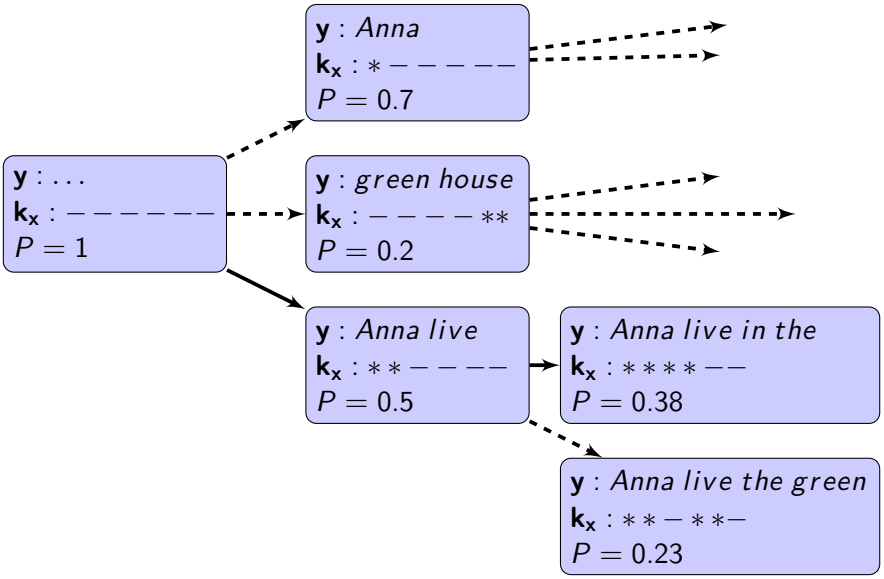


Figure 1.2. Decoding procedure for the example sentence $\mathbf{x}$ ="*Ana vive en la casa verde .*". In this figure, the vector $\mathbf{k_x}$ represents which words of the source sentence $\mathbf{x}$ have been translated until that point, the character $-$ represents the word $\mathbf{x}_i$ has not yet been translated, and $*$ indicates that word has been translated. The probability P of each hypothesis is given only for illustrative purposes.

Figure 1.2 shows the procedure for translating a source sentence $\mathbf{x}$ ="*Ana vive en la casa verde .*", following the phrases extracted in Figure 1.1. This example illustrates the decoding algorithm described in [32], where the translation is generated sequentially from left to right and re-ordered. In this figure, the initial (empty) hypothesis is first expanded into several partial hypotheses by using different phrases. The use of these phrases leads to different coverage vectors denoted by the vector $\mathbf{x}$, indicating which words of the source sentence $\mathbf{x}$ have been translated until that point. Character $-$ denotes that word $\mathbf{x}_i$ has not been translated yet, and $*$ indicates that the word has been translated. The hypothesis probability is computed as a product. For this reason, translating more source words leads to a lower probability mass being assigned to that specific hypothesis. Since hypothesis expansion is done by expanding first those hypotheses with the highest probabilities, the algorithm would keep

expanding hypotheses with fewer translated words. The algorithm copes with this problem using the coverage vector. It allows to compete among each other only those hypotheses with the same amount of translated words. For instance, in the second expansion in Figure 1.2, the hypotheses that would compete among each other would be **green house** and **Anna live**. In this figure, the probability P of each hypothesis is given only for illustrative purposes.

1.3.2 LANGUAGE MODELS

Language models (LM) are a crucial part of SMT [5]. A statistical language model includes the language regularities in a probabilistic way. Higher probabilities are given to common sequences of words whereas lower probabilities are given to unseen words. This probability distribution $p(\mathbf{x})$ tries to reflect how frequently does the string $\mathbf{x}$ appears in the whole text.

The n-gram model, perhaps the most widespread statistical language model, was proposed by [34] and has proved to be robust and effective. A n-gram is a contiguous sequence of n words from a given corpus or document and n is the order of the n-gram. We will introduce the n-gram language models considering that $n = 2$; these models are the so-called bi-gram language models. Let us consider the sentence $\mathbf{x}$ composed of the words $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_J\}$, we can express $p(\mathbf{x})$ as follows:

$$\begin{aligned} p(\mathbf{x}) &= p(x_1) \dots p(x_2|x_1) \dots\dots p(x_J|x_1 \dots x_{J-1}) \\ &= \prod_{j=1}^{J} p(x_j|x_1 \dots x_{j-1}) \end{aligned} \tag{1.8}$$

Bigram models assume that the probability of a given word depends only on the immediately preceding word:

$$p(\mathbf{x}) = \prod_{j=1}^{J} p(x_j|x_1, \dots, x_{j-1}) \approx \prod_{j=1}^{J} p(x_j|x_{j-1}) \tag{1.9}$$

To estimate the probability $p(x_j|x_{j-1})$, the frequency of the word x_j:

$$p(x_j|x_{j-1}) = \frac{c(x_{j-1}x_j)}{\sum_{x_j} c(x_{j-1}x_j)} \tag{1.10}$$

Here, $x_{j-1}x_j$ is a concatenation between the words x_{j-1} and x_j, $c(x_j)$ is the count of occurrences of the word x_j in the corpus or document at hand. When the degree of the n-grams considered is $n > 2$, we condition the probability of the words to $n-1$. The probability of a sentence is calculated as follows for $n > 2$:

$$p(\mathbf{x}) = \prod_{j=1}^{J} p(x_j | x_{j-(n-1)}, ..., x_{j-1}) \tag{1.11}$$

In this case, the probability of a given n-gram is very similar to the one described in Equation 1.10 for the case of bi-gram model:

$$p\left(x_j | x_{j-(n-1)}^{j-1}\right) = \frac{c\left(x_{j-(n-1)}^{j-1}\right)}{\sum_{x_j} c\left(x_{j-(n-1)}^{j-1}\right)} \tag{1.12}$$

Here $x_{j-(n-1)}^{j-1}$ denotes the segment of the sentence $\mathbf{x}$ which starts at the word $x_{j-(n-1)}$ and finishes at the word x_{j-1}. According to what is described in the literature, the value of n is typically set to 5 in the case of SMT.

1.3.2.1 LM Smoothing

Considering that most words are uncommon, statistically speaking, a large number of n-grams are unlikely to appear in a particular corpus. In Equation 1.12, a probability of zero is assigned to these unseen events even though they are linguistically valid sequences. Due to the overall probability of a sentence being calculated as the product of the probabilities of each subsequence (up to the order of n) it is composed of, any zero-probability subsequence will produce a probability estimate of zero for the sentence. The solution to this problem are the smoothing techniques. These techniques can be used to improve the estimated probabilities in a language model for sparse or unseen n-grams.

An extensive survey of the many smoothing techniques that have been developed in the thirty years of statistical language modelling research is given by [35]. A wide variety of smoothing techniques exist. Nevertheless, the technique most used in SMT was proposed by [36].

1.3.2.2 LM Evaluation

In order to evaluate the performance of the language model, different scores have been proposed in the literature. The simplest one is the average log-likelihood of the test samples [37]. This metric can be seen as an empirical estimation of the cross-entropy. The cross-entropy of a corpus S with a probability distribution P according to another distribution Q is:

$$H_S(P,Q) = -\sum_{s \in S} P(s)\, logQ(s) \tag{1.13}$$

Cross-entropy is the basis of perplexity, which is used to assess language model performance [38]. Perplexity measures the probability that a language model assigns to a sample test. It is defined as:

$$PP = 2^{-\frac{1}{n}\sum_{i=1}^{n} logp_{LM}(w_i|w_1,\ldots,w_{i-1})} \tag{1.14}$$

1.4 Continouns vector-space representation

The idea of representing words in a vector-space using neuronal networks was originally proposed by [39–41]. Building continouns vector-space representations (CVR) of words/sentences, or word embeddings, has always generated much interest in the NLP community. CVR of words have been widely used in a variety of NLP applications. These representations have recently demonstrated promising results across a variety of tasks [42–46], such as speech recognition, part-of-speech tagging, sentiment classification, information retrieval, identification and machine translation.

The limitation of the original proposals, back in the 1980s, was that computational requirements quickly became unpractical for growing vocabulary sizes $|V|$. However, work performed recently in [47–52] made it possible to overcome such a drawback, while still relying on neural network language models, in which words are represented as high dimensional real valued vectors.

In 2003, Bengio *et. al.*, [47] introduced feed forward neural networks into traditional n-gram language models, which might be the foundationalwork of neural network language models. In this model,

words were represented by a low-dimensional vector and the parameters could be learned using unsupervised methods. Later on, in 2008 [53], the authors proposed a unified neural network architecture to learn word embeddings instead of the time consuming softmax layer improving the training speed significantly. [54] reduced the computational complexity of Bengio's model by replacing the softmax layer with a tree structured probability distribution.

State-of-the-art word embeddings models were used in [49,52,55], by removing the hidden layers of the neural network and proposing two new models: the Continuous Bag of Words Model and the Continuous Skip-Gram Model. In Section 1.4.1, we explain in more detail these two models.

Several multi-prototype models have been proposed to alleviate the problem caused by the polysemy and homonym. A word is polysemous if it can be used to express different meanings. For instance, the word "bank" cannot have high cosine similarity with the words "river" and "money" at the same time since these two words are so dissimilar, thus, the single vector representation of word "bank" cannot express two different meanings. The homonym problem appears when two or more words sound the same (homophones), have the same spelling (homographs), or both, but do not have related meanings. For example, [56] took advantage of bilingual resources and the affinity propagation clustering algorithm to learn multiple embeddings corresponding to multiple word senses, because of a polysemous word in one language could not be exactly a polysemous word in another language. Following this line, [57] pre-clustered the corpus into specified classes and relabelled the tokens into different classes. Then, specific numbers of embeddings were learned per word type. In addition, [58] shifted clusters into the training process and proposed a non-parametric clustering model which could dynamically generate new clusters based on word meaning. Finally, [59] proposed a supervised fine tuning framework to transform the existing single-prototype word embeddings into multi-prototype word embeddings based on lexical semantic resources.

1.4.1 WORD EMBEDINGS

Word embeddings models have the purpose to map words with similar meanings to similar vectors. The basic idea is to represent each word in the vocabulary V, with a real-valued vector of some fixed dimension *size* , capturing the similarity (lexical, semantic and syntactic) between the words.

Two approaches were proposed by [49], namely, the Continuous Bag of Words Model (CBOW) and the Continuous Skip-Gram Model (Skip-Gram) . CBOW forces the neural net to predict the current word using the surrounding words whereas Skip-Gram forces the neural net to predict surrounding words using the current word. These two approaches were compared to previously existing approaches, such as the Feed-forward Neural Net Language model proposed in [47], and the Recurrent Neural Net Language model [48], obtaining considerably better performance in terms of training time in semantic and syntactic word relationship tasks.

1.4.2 CONTINUOUS SKIP-GRAM MODEL

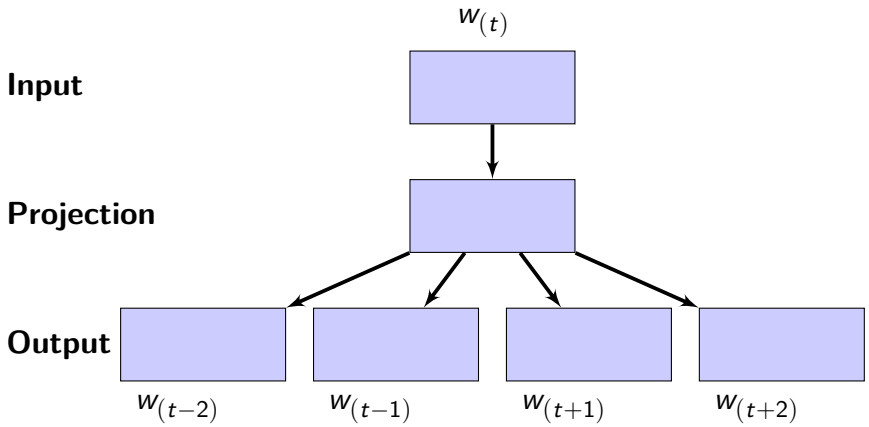


Figure 1.3. Skip-gram model architecture. The objective is to learn word vector representations within a certain range before and after the current word.

The use of log-linear models has been proposed [49] as an efficient way to generate representations of words, since they reduce the complexity of the hidden layer thereby improving efficiency. In contrast

to the CBOW model, the Skip-Gram uses word ordering to sample distant words that appear less frequently during training time. Experimental results also demonstrated that this model offers better performance on average, excelling especially at the semantic level. These results were confirmed in our own preliminary work and hence, we used the Skip-Gram approach to generate our distributed representations of words.

The continuous Skip-Gram model [49, 55, 60] is an iterative algorithm. Figure 1.3 shows a graphical representation of the model, which attempts to maximize the classification of the context surrounding a word. In Skip-Gram model we are given a corpus of words w and their contexts c. It consider the conditional probabilities $p(c|w)$, and given a corpus T, the goal is to set the parameters θ of $p(c|w,\theta)$ so as to maximize the corpus probability:

$$\underset{\theta}{\operatorname{argmax}} \prod_{(w,c)\in D} p(c|w,\theta) \tag{1.15}$$

Here, D is the set of all word and context pairs we extract from the text. This model requires the formulation of $p(c|w,\theta)$ using the softmax function, which is given by:

$$p(c|w,\theta) = \frac{\exp^{v_c \cdot v_w}}{\sum_{c' \in C} \exp^{v_{c'} \cdot v_w}} \tag{1.16}$$

where v_c and $v_w \in \mathbb{R}^d$ are vector representations for c and w respectively, and C is the set of all available contexts. The parameters θ are v_{c_i}, v_{w_i} for $w \in V$, $c \in C$ and $i \in 1, \cdots, d$. Now, taking the Equation1.15 and switch from product to sum:

$$\underset{\theta}{\operatorname{argmax}} \sum_{(w,c)\in D} log p(c|w,\theta) = \sum_{(w,c)\in D} \left(log \exp^{v_c \cdot v_w} - log \sum_{c'} \exp^{v_{c'} \cdot v_w} \right) \tag{1.17}$$

1.4.2.1 *Negative Sampling*

A computationally efficient approximation of the full softmax function is called negative sampling, and was introduced by [55]. The negative sampling function is a simplified version of the Noise Contrastive Estimation (NCE) [54, 61], which is only concerned with preserving vector quality in the context of Skip-gram learning.

The basic idea is to use logistic regression to distinguish the target word W_O from a noise distribution $Pn(w)$, having k negative samples for each word. Formally, negative sampling estimates $p(w_O|w_I)$ as follows:

$$log\ \sigma({v'_{w_O}}^T v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[log\ \sigma(-{v'_{w_i}}^T v_{w_I}) \right] \tag{1.18}$$

which is used to replace every $logP(w_O|w_I)$ term in the Skip-gram objective. Note that computational complexity is linear with the number of negative samples k. The experimental results in [55] show that this function obtains better results at the semantic level than hierarchical softmax and classical NCE. Therefore, in this thesis we will use negative sampling in all our experiments.

1.4.3 SENTENCE EMBEDDINGS METHODS

A problem that arises when using CVR of words is how to represent a whole sentence (or document) with a continuous vector. Following the idiosyncrasy described in the previous section (i.e., semantically close words are also close in their CVR), we present in this section the different sentence representations used in the present work.

Numerous works have attempted to extend the CVR of words to the sentence or phrase level (just to name a few, [44, 62–65]). In this thesis, we used two different CVRs of sentences, which have been called `Mean-vec` and `Document-vec` .

However, first it is necessary to introduce some definitions. Let $f(w)$ be the word embeddings representation of word w, included in sentence $\mathbf{x}$. We will further denote as the CVR of given sentence $\mathbf{x}$ as $F(\mathbf{x})$. In some cases we will denote $F(\mathbf{x})$ as $F_{\mathbf{x}}$ to simplify notation. Then, we define the CVR of sentences as:

Mean-vec: It is the most simple and common approach for representing a vector of a sentence by summing or averaging the word embeddings participating in the sentence. In our work, we used a weighted arithmetic mean of all the words in the document or sen-

tence (as proposed by [55, 66] :

$$F_{\mathbf{x}} = F(\mathbf{x}) = \frac{\sum_{w \in \mathbf{x}} N_{\mathbf{x}}(w) f(w)}{\sum_{w \in \mathbf{x}} N_{\mathbf{x}}(w)} \tag{1.19}$$

where x is a word that appears in sentence $\mathbf{x}$, $f(w)$ is the CVR of w, obtained as described above in Section 1.4.1, and $N_{\mathbf{x}}$ is the count of w in sentence $\mathbf{x}$.

Document-vec: A more sophisticated approach is presented by [64]. The authors adapted the continuous Skip-Gram model [55] (Section 1.4.2) to generate representative vectors of sentences or documents. *Document-vec* [1] follows the Skip-Gram architecture to train a special vector $F_{\mathbf{x}}$ representing the sentence or document. Basically, before each context window movement, the idea is to use $F(\mathbf{x})$ in place of $f(w)$, with the objective of maximizing the classification of the surrounding words. The context window can be defined as the window of words to the left and to the right of the target word.

Another approach to obtain sentence embeddings is based on the encoder-decoder architecture [65, 67–69]. In this method, an encoder network is used to produce a vector representation of the sentence, which is then fed as input into a decoder network that uses it to perform some prediction task (i.e. recreate the sentence, or produce a translation of it). The encoder and decoder networks are trained jointly in order to perform the final task.

1.5 Neural Statistical Machine Translation

NMT is a newly emerging approach to SMT, which has made promising progress in recent years [45, 67, 68, 70–77].

As opposed to the PB model, NMT implements a neural network that directly models the conditional probability attempting to build and train a large neural network that reads a sentence and outputs a correct translation. Hence, the decomposed conditional probability

[1] http://radimrehurek.com/gensim/models/doc2vec

$p(\mathbf{y}|\mathbf{x})$ is:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{I} p(y_i|y_1^{i-1}, \mathbf{x}) \tag{1.20}$$

where the model is often trained to predict the next word y_i, given $\mathbf{x}$ and all words $y_1, \ldots, y_{i-1}$.

Most of the proposed neural machine translation models belong to the family of **encoder–decoder** models (see Figure 1.4) [67]. The "encoder-decoder" name comes from the idea that the first neural network running over "encodes" its information as a vector of real valued numbers (the hidden state), then the second neural network is used to predict "decodes" this information into the target sentence. In Section 1.5.1, we explain the encoder-decoder architecture used in this work.

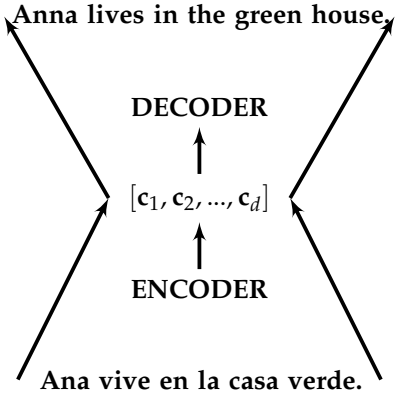


Figure 1.4. Encoder-decoder model for Neural Machine Translation.

1.5.1 ENCODER-DECODER ARCHITECTURE

Figure 1.5 shows an example of the first encoder-decoder architecture, presented in [65, 67]. This model aims to solve the mapping of a sequence to another sequence, for sequences with arbitrary lengths. The source sequence $\mathbf{x} = x_1, \ldots, x_J$ is encoded into a vector via an encoder, which is then decoded to a target sequence $\mathbf{y} = y_1, \ldots, y_I$ via a decoder by maximizing the predictive probability. Both the encoder and the decoder are typically implemented via a Recurrent Neural

Network (RNN) , although there is no restriction on which particular type of neural network is used as either an encoder or a decoder.

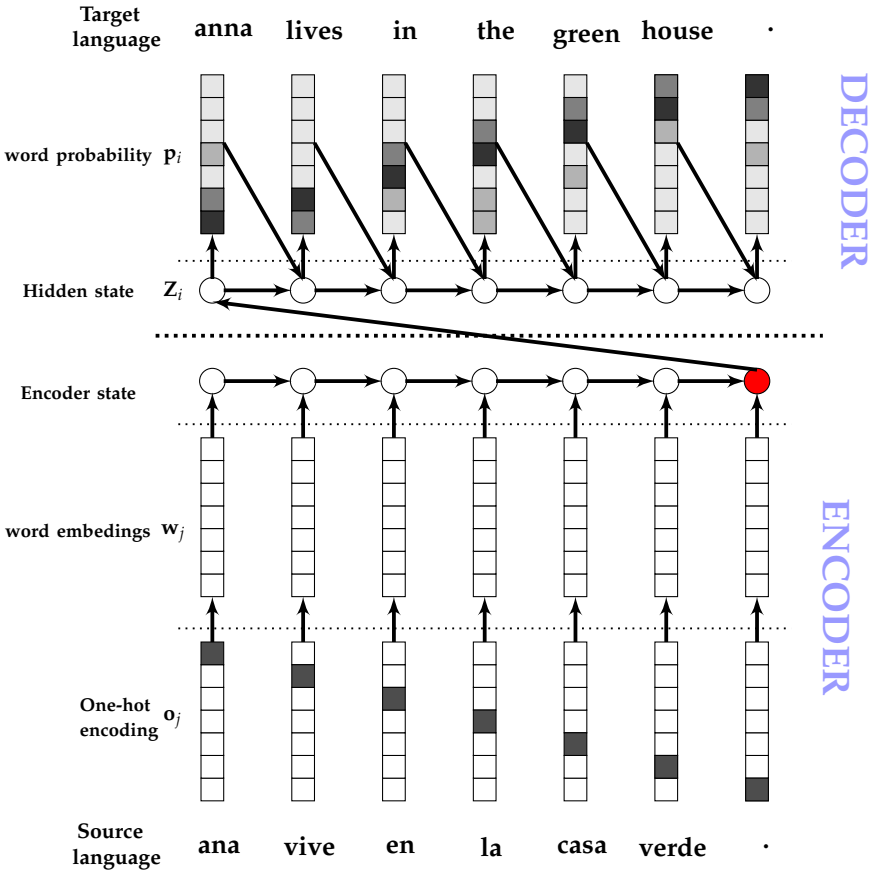


Figure 1.5. Encoder-decoder architecture for Neural Machine Translation.

Encoder

Typically, there are three steps for encoding a source sentence:

1. One-hot vector representation of a word: Each word x_j in the source sentence $\mathbf{x}$ is represented as a vector $\mathbf{o}_j \in \{0,1\}^{|V|}$, where $\mathbf{o}_j$ has the same dimensionality as the size of the dictionary, i.e.,

$|V|$ and has an element of one at the location corresponding to the location of the word in the dictionary and zero elsewhere.

2. Word embeddings: Words are transformed into a representation in the low dimensional semantic space. This representation is explained in detail in Section 1.4.

3. Encoding of the source sequence via RNN: This can be described mathematically as:

$$\mathbf{h}_j = \phi_\theta(\mathbf{h}_{j-1}, \mathbf{w}_j) \tag{1.21}$$

where h_0 is a zero vector, ϕ_θ is a non-linear activation function, and $\mathbf{h} = \{\mathbf{h}_1, \ldots, \mathbf{h}_J\}$ is the sequential encoding of the first J words from the source sequence. The vector of the whole sentence $\mathbf{x}$ can be represented as the encoding vector at the last time step J with $\mathbf{h}$. For a more sophisticated sentence encoding, we can use a Bi-directional RNN. This can be implemented using bi-directional Gated Recurrent Units (GRU) [78].

Decoder

The goal of the decoder is to obtain an output sentence employed the encoder output. The typical steps for decoding are:

1. At each time step i, given the encoding vector of the source sentence, the i-th word u_i from the target language and the RNN hidden state z_i, the next hidden state z_{i+1} is computed as:

$$\mathbf{z}_i = \phi(\mathbf{z}_{i-1}, u_{i-1}, \mathbf{c}_i) \tag{1.22}$$

where ϕ is a function with various choices, such as a feed-forward layer, a GRU, a LSTM, etc. and u_{i-1} denotes the embedding of the previous output word. $\mathbf{c}_i$ is the input context (which we still have to define in next section).

2. Calculate the probability p_{i+1} for the $i+1$-th word in the target language sequence by normalizing z_{i+1} using softmax.

3. Repeat Steps, until all the words in the target language sentence have been processed.

Attention mechanism

In the encoding stage, several problems appear with a fixed dimensional vector representation:

- It is very challenging to encode both the semantic and syntactic information of a sentence with a fixed dimensional vector regardless of the length of the sentence.
- The other problem is in terms of attention. Intuitively, when translating a sentence, we typically pay more attention to the parts in the source sentence more relevant to the current word being translated. Moreover, the focus changes along the process of translation. However, when using a fixed dimensional vector all the information from the source sentence is treated equally, in terms of attention for all words being translated.

Therefore, [68] introduced the attention mechanism, which can decode based on different fragments of the context sequence in order to address the difficulty of feature learning for long sentences. At each decoding time-step i, the attention mechanism computes a different context vector $\mathbf{c}_i$ as the weighted sum of the sequence of hidden states $\mathbf{h}_J$ from the encoder:

$$\mathbf{c}_i = \sum_{j=1}^{J} \alpha_{ij} \mathbf{a}_j \tag{1.23}$$

where α_{ij} denotes the weight assigned to each $\mathbf{a}_j$. This weight is the strength of attention of the i-th word in the target language sentence to the j-th word in the source sentence. Figure 1.6 shows the encoder-decoder architecture of an attention-based NMT system. The weight a_{ij} is calculated as:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{j=1}^{J} exp(e_{ij})} \tag{1.24}$$

where these weights can be interpreted as the alignment score between target and source tokens. More precisely, the fitness is computed with the i-th hidden state $\mathbf{z}_i$ of the decoder RNN and the j-th

context vector $\mathbf{a}_j$ of the source sentence. In an attention model, every word in the source sentence is related to every word in the target language sentence. The strength of the relation is a real number computed via the model and thus, can be trained via back-propagation. The feed-forward network for the attentional weights has been computed in 1.25

$$e_{ij} = \mathbf{v}_{\mathbf{a}^T} \, tanh(\mathbf{W}_D h'_j + \mathbf{W}_E \mathbf{a}_j) \tag{1.25}$$

where $\mathbf{W}_D$ and $\mathbf{W}_E$ represent the weight matrices transformations from the decoder and encoder, respectively and $\mathbf{a}$ corresponds to a weight vector.

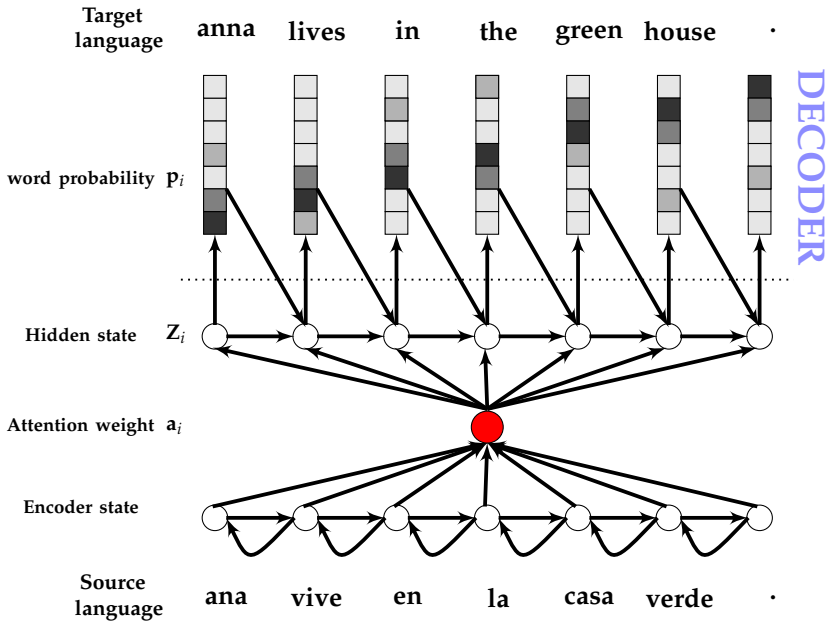


Figure 1.6. Decoder with attention mechanism for Neural Machine Translation.

1.5.2 TRAINING

Once we have both, a model configuration and a training corpus, the training process starts. One problem appears with the number of steps variation between the encoder and decoder in each training

example. Training goal is based on the probability mass given to the correct word, given a perfect context.

Finally, practical training of NMT systems requires GPUs which are well suited to the high degree of parallelism inherent in these deep learning models. To increase parallelism even more, we process several sentence pairs at once.

1.5.3 Decoding with beam search

Once the model parameters are estimated, the goal of the NMT system is the same as in the case of PBSMT. An optimal solution would require to search over the space of all possible target sentences, which is in practice unaffordable. To generate translations, suboptimal decoding strategies such as beam search were used by [45, 67, 68]. Beam search is a commonly used decoding technique that improves translation performance in neural machine translation systems [79]. Instead of decoding the most probable word in a greedy fashion, beam search keeps several hypotheses (or "beams") in memory and finally chooses the best one based on a scoring function.

1.6 Summary

In this chapter we have introduced the field of MT. We have paid special attention to the SMT approach since it is the one in which this thesis is focused on. We have described the main approaches to define the statistical models involved in the translation process. This includes the n-gram language models and the log-lineal models with emphasis in phrases-based models. An introduction to neuronal machine translation using the encoder-decoder architecture is also given. The concept of word embeddings was also introduced, as well as different techniques that will be used to represent words or sentences within a continuous vector space.

2 SMT experimental framework

* * *

"Por las mañanas
Mi pequeñuelo
Me despertaba
Con un gran beso."

—JOSÉ MARTÍ
MI CABALLERO

"In the mornings
My little guy was
I woke up
With a big kiss."

—MICROSOFT TRANSLATOR
MY KNIGHT

* * *

2.1 Introduction

This chapter presents the experimental framework used for this thesis. Different automatic evaluation criteria used to evaluate the translation quality are introduced in Section 2.2. The experiments in this work were made using several domains and language pairs, each of which is detailed in Section 2.3. To enable direct comparisons between the methods we explored, we use one common framework for all the experiments presented. For this reason, the principal toolkit employed is presented in Section 2.4.

Table 2.1 shows the abbreviations introduced in the current chapter, in order to facilitate a better comprehension of the text.

Table 2.1. Abbreviations used in Chapter 2.

Abbreviation	Description
BLEU	Bilingual Evaluation Understudy metric
TER	Translation Edit Rate metric
METEOR	METEOR automatic metric
Moses	Moses SMT toolkit
GIZA	word alignment toolkit
SRILM	language modelling toolkit
NMT-Keras	NMT toolkit
word2vec	word embeddings toolkit

2.2 Evaluation criteria

Evaluation of SMT translated output would ideally be judged by human evaluators. However, this is subjective and costly for experimental purposes. With this situation, machine translation research relies on having automatic methods for evaluating translations. This process is easier for sentences which already have human translations to compare against, and such translation is typically called *reference*. In this section we will review the automatic evaluation measures that are commonly used in SMT, emphasizing those that will be used to evaluate the techniques proposed in this thesis.

Specifically, the SMT evaluation metrics used are Bilingual Evaluation Understudy (BLEU) [80], METEOR [81, 82] and Translation Edit Rate (TER) [83] , which are the most popular evaluation metrics used in MT.

2.2.1 Bilingual Evaluation Understudy

This score [80] measures the precision of unigrams, bigrams, trigrams, and fourgrams with respect to a set of reference translations with a penalty for too short sentences. This metric is not an error rate, which means that the higher score, the better. BLEU will be reported as a

percentage, ranging from 0 to 100.

$$\textbf{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{4} \frac{\log p_n}{N}\right) \tag{2.1}$$

Here BP is a penalisation factor, p_n denotes the precision of n-grams in the hypothesis translation, $n = 4$ is typically used.

2.2.2 METEOR METRIC

This metric [81, 82] is an automatic evaluation metric that scores machine translation hypotheses by aligning them to one or more reference translations. It is based on alignments and they rely on exact, stem, synonym, and paraphrase matches between words and phrases. Segment and system level metric scores are calculated based on the alignments between hypothesis-reference pairs.

2.2.3 TRANSLATION EDIT RATE METRIC

This score [83] is an error metric that measures the minimum number of edits required to change a translation obtained by the system into the reference. TER value is obtained as the minimum number of edits (*#edits*) required to modify the system translation so that it matches the reference translation, normalised by the average number of reference words $|\hat{\mathbf{y}}|$.

$$\textbf{TER} = \frac{\#edits}{|\hat{\mathbf{y}}|} \tag{2.2}$$

The different edits are insertions, deletions, substitutions of single words and shifts of word sequences. Usually, TER is reported as a percentage, although it can yield values over 100.

In addition to the metric results, confidence interval sizes will be also provided with the purpose of assessing whether differences in BLEU, TER and METEOR are statistically significant or not. To this end, all PBSMT results reported in this thesis are the outcome of the average of 10 repetitions of the tuning process with 95% confidence intervals. In the case of NMT, we use an efficient implementation of the method described in [84], where the goal is to control the computational and time cost. The fundamental idea of applying such implementation relies in performing the bootstrap re-sampling on the

sentence-level counts which lead to the translation scores used, and not on the sentences as such. Hence, much computational effort is saved, since it is not needed to translate b test sets, obtain such counts, and then compute the final translation quality scores. The only thing needed is to repeat the computation of the final scores b times. For this reason, obtaining the confidence intervals ends up being very cheap. The confidence intervals reported in this thesis were obtained after performing $b = 10.000$ bootstrap re-sampling repetitions unless stated otherwise.

2.3 Corpora

This section describes the different parallel corpora that will be used to test the techniques proposed in this thesis. All SMT experiments require usually three corpora: the training, the development and the test corpora. The training corpus is used to train SMT models, the test corpus is used to obtain quality measures as those defined in Section 2.2 whereas the development corpus is used to adjust specific parameters of the statistical models such as the log-linear combination weights described in Section 1.3.

In addition, let us define some notations which will be useful for the remainder of the text: M denotes millions of elements and k thousands of elements, $|S|$ stands for the number of sentences, $|W|$ for the number of words and $|V|$ is the vocabulary size given the corpus vocabulary V. Finally, other smaller corpora will be also used for the purpose of evaluating the techniques described in some specific chapters. Given that, these corpora will be only used on specific occasions, thus, their description will be provided when appropriate.

EUROPARL CORPUS

The Europarl[1] corpus [85] is extracted from the proceedings of the European Parliament, which are written in all the languages of the European Union. Specifically, the experiments conducted on this thesis refer to version 7 of the Europarl corpus. It was used in the shared task

[1] www.statmt.org/europarl/

of the Workshop on Statistical Machine Translation [2]. Table 2.2 shows some statistics of this corpus, which includes parallel texts from four European language pairs, specifically English-French, English-German, English-Spanish and English-Czech.

Table 2.2. Europarl corpus main features. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

		EN-FR		
Domain	Corpus	$\|S\|$	$\|W\|$	$\|V\|$
Europarl	Train	2.0M	50.2M - 52.5M	157.7k - 215.2k
	Dev	2.0k	40.8k - 48.6k	5.1k - 6.2k
		EN-DE		
Europarl	Train	1.9M	47.8M - 44.6M	153.4k - 290.8k
	Dev	2.0k	49.8k - 46.4k	8.6k - 10.9k
		EN-ES		
Europarl	Train	1.9M	49.1M - 51.6M	308.9k - 422.6k
		EN-CS		
Europarl	Train	537k	11.6M - 10.0M	60.6k - 164.3

HANSARDS CORPUS

The Canadian Hansard corpus [86,87] consists of a set of aligned texts in French and English languages. These texts are extracted from the official records of the Canadian Parliament. Main features of this corpus are shown in Table 2.3.

EMEA CORPUS

The EMEA corpus [88] is available in 22 languages and contains documents from the European Medicines Agency. The development and test corpora for the medical domain are the partitions established in the Workshop on Statistical Machine Translation (WMT) [89] of the Association for Computational Linguistics, in 2014. Table 2.4 shows

[2] www.statmt.org/wmt16/

Table 2.3. Hansard corpus main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

	EN-FR		
Corpus	$\|S\|$	$\|W\|$	$\|V\|$
Hansard	8.1M	144.4M - 161.6M	186.7k - 191.2k

some statistics of this corpus. It includes parallel texts from European language pairs, specifically English-French and English-German.

Table 2.4. EMEA corpora main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

		EN-FR		
Domain	Corpus	$\|S\|$	$\|W\|$	$\|V\|$
EMEA	Train	1.0M	12.1M - 14.1M	98.1k - 112k
	Dev	0.5k	9.8k - 11.6k	0.9k - 1.0k
	Test	1.0k	21.4k - 26.9k	1.8k - 1.9k
		DE-EN		
EMEA	Train	1.1M	10.9M - 12.9M	141k - 98.8k
	Dev	0.5k	8.6k - 9.2k	0.8k - 0.9k
	Test	1.0k	18.2k - 19.2k	1.7k - 1.9k

NEWS CORPUS

Another corpus that will be used in some chapters of this thesis is the News-Commentary (NC) corpus [90]. The NC corpus is composed of translations of news articles. Characteristics are provided in Table 2.5.

INFORMATION TECHNOLOGY CORPUS

This corpus has been obtained for the Information Technology (IT) adaptation [3] task in the First Conference on Machine Translation [91].

[3] http://www.statmt.org/wmt16/it-translation-task.html

Table 2.5. NC corpora main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

		EN-FR		
Domain	Corpus	$\|S\|$	$\|W\|$	$\|V\|$
NC	Train	157k	3.5M - 4.0M	65.1k - 76.7k
	Dev	1.5k	24.1k - 24.9k	2.04k - 2.1k
	Test	1.5k	23.6k - 25.9k	2.0k - 2.1k
		DE-EN		
NC	Train	178k	4.0M - 3.9M	98.4k - 70.0k
	Dev	2.2k	38.1k - 40.7k	3.4k - 3.7k
	Test	3.0k	53.9k - 56.7k	4.7k - 4.9k

The IT translation task is focused on domain adaptation of MT to the information technology domain and translation of answers in a cross-lingual help-desk service. Hardware and software troubleshooting answers are translated from English to the user's languages: Bulgarian, Czech, German, Spanish, Basque, Dutch and Portuguese. Data sets were created within the QTLeap project [4]. Characteristics are provided in Table 2.6.

Table 2.6. IT corpora main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

		EN-ES		
Domain	Corpus	$\|S\|$	$\|W\|$	$\|V\|$
IT	Train	157k	865.1k - 977.7k	119.7k - 128.8k
	Dev	2.0k	39.0k - 41.9k	6.3k - 6.9k
	Test	1.0k	18.5k - 21.2k	3.8k - 4.2k
		EN-CS		
IT	Train	132.7k	766.6k - 666.7k	112.4k -133.1k
	Dev	2.0k	39.0k - 34.2k	6.3k - 8.1k
	Test	1.0k	18.5k - 16.8k	3.8k - 5.0k

[4] http://qtleap.eu/

XEROX CORPUS

The Xerox corpus (XRCE) [92] consists of translations of Xerox printer manuals involving four different language: English, French, Spanish, and German. The main features of these corpora are shown in Table 2.7.

Table 2.7. XRCE corpora main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

		EN-FR		
Domain	Corpus	$\|S\|$	$\|W\|$	$\|V\|$
XRCE	Dev	976	11.9k - 13.2k	6.3k - 1.2k
	Test	936	11.4k - 12.3k	1.1k - 1.2k
		EN-ES		
XRCE	Dev	1.0k	14.2k - 15.9k	4.2k - 3.8k
	Test	1.1k	8.4k - 10.1k	1.6k - 1.7k
		DE-EN		
XRCE	Dev	964	11.0k - 11.1k	1.4k - 1.1k
	Test	995	12.4k - 12.6k	1.5k - 1.1k

COMMON CRAWL CORPUS

The Common Crawl (COMMON) corpus [93] which was collected from web sources. The main features of this corpus are shown in Table 2.8.

Table 2.8. Common Crawl corpus main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

	EN-ES		
Corpus	$\|S\|$	$\|W\|$	$\|V\|$
COMMON	1.8M	40.7M - 43.5M	530.5k - 613.8k

UNITED NATION CORPUS

The United Nations (UN) corpus [94] consists on United Nations official records and other parliamentary documents belonging to the public domain. The main features of this corpus are shown in Table 2.9.

Table 2.9. UN corpus main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

	EN-ES		
Corpus	$\|S\|$	$\|W\|$	$\|V\|$
UN	11.2M	280.7M - 327.3M	801.4k - 893.2k

ONE BILLION WORD CORPUS

The One Billion Words corpus [95] is a monolingual English corpus created for measuring progress in statistical language modelling. The main features of this corpus are shown in Table 2.10.

Table 2.10. One Billion Word corpus main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

	EN		
Corpus	$\|S\|$	$\|W\|$	$\|V\|$
1 Billion Words	30.3M	800M	700k

ELECTRONIC COMMERCE CORPUS

This corpus was obtained from a real e-commence page *Cachitos de Plata* [5]. Statistics of this corpus is provided in Table 2.11

Table 2.11. Real e-Commerce corpus main figures. M stands for millions and k thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

		EN-ES		
Domain	Corpus	$\|S\|$	$\|W\|$	$\|V\|$
e-Commerce	Test	886	7.3k - 8.6k	558 - 588

Table 2.12. Development and Test corpora from the Johns Hopkins adaptation corpora. M stands for millions and k for thousands of elements; $|S|$ stands for the number of sentences, $|W|$ stands for the number of words and $|V|$ for vocabulary size.

		EN-FR		
Domain	Corpus	$\|S\|$	$\|W\|$	$\|V\|$
EMEA	Dev-in	2.0k	27.8k - 31.8k	2.2k - 2.5k
	Test	2.0k	24.6k - 29.0k	2.1k - 2.3k
NEWS	Dev-in	2.0k	49.4k - 55.3k	3.7k - 3.8k
	Test	2.5k	61.6k - 69.4k	4.4k - 4.8k
PRESS	Dev-in	2.0k	52.0k - 64.9k	4.2k - 4.6k
	Test	1.9k	52.2k - 65.0k	4.3k - 4.6k
SUBS	Dev-in	3.0k	32.2k - 29.5k	1.8k - 1.9k
	Test	3.3k	36.0k - 31.9k	2.0k - 2.0k

JOHNS HOPKINS ADAPTATION CORPORA

These corpora were employed by the domain adaptation task taken from the Johns Hopkins Summer Workshop 2012 [87]. Corpora come from four different domains: the medical domain (referred to as EMEA), the general news domain (referred to as NEWS), the press domain (referred to as PRESS), and the subtitle domain (referred to as SUBS). Statistics are provided in Tables 2.12.

2.4 Toolkits

This section describes the different NLP toolkits that have been used in this thesis. Principal PBSMT toolkits are: the SMT toolkit Moses,

[5] http://www.cachitosdeplata.com

the word-alignment toolkit Giza++ and the language modelling toolkit SRILM.

Moses SMT toolkit: Moses [96] is an open source SMT toolkit, licensed under the LGPL license. It includes a large amount of tools to train and optimise PB SMT systems, as well as a decoder for translating source texts using the models built.

GIZA++ word-alignment toolkit GIZA++ [15] is a SMT toolkit that implements training and search for IBM models 1-5 and HMM. The GIZA++ toolkit is used to build the word-alignments which are the most important steps when a phrase-table is created.

SRI Language Modelling toolkit SRILM [97] is a toolkit for building language models and is under development since 1995 by the Stanford Research Institute (SRI) Speech Technology and Research Laboratory. The SRILM toolkit provides a wide range of libraries and scripts which can be used independently in different tasks.

Clean and tokenizer tools Corpora are cleaned and tokenised employing the tools available in the Moses toolkit.

Automatic quality scores The metrics were computed using the scripts provided in the Moses toolkit page.

Neural machine translation toolkit We used the NMT-Keras [6] [98] toolkit for building the NMT systems described in the Section 1.5. The toolkit offers scalable training and inference for the most prominent encoder-decoder architectures: attentional recurrent neural networks, conditional GRU with attention, etc..

Word embeddings toolkit The Word2vec toolkit [7] provides an efficient implementation of different neural architectures for computing vector representations of words. In this toolkit, the implemented

[6] https://github.com/lvapeab/nmt-keras

[7] https://code.google.com/archive/p/word2vec

methods are the Continuous Bag of Words Model and the Continuous Skip-Gram Model, proposed by [49]. These representations can be used in many natural language processing applications.

2.5 Summary

In this chapter, we have described the parallel corpora as well as the evaluation measures and toolkits that will be used in the following chapters to test the techniques proposed in this thesis.

3 Data selection preliminaries

* * *

"Having faith in God did not mean sitting back and doing nothing. It meant believing you would find success if you did your best honestly and energetically."

—KEN FOLLETT
THE PILLARS OF THE EARTH

"Tener fe en Dios no significaba sentarse y no hacer nada. Significaba creer que tendrías éxito si lo hicieras con honestidad y energía."

—GOOGLE TRANSLATOR
LOS PILARES DE LA TIERRA

* * *

3.1 Introduction

This chapter is an advance of the historical and current research landscape related to our application of data selection methods in SMT. The first part presents the adaptation problem in SMT and the state-of-the-art adaptation techniques on different SMT paradigms. We pay special attention to the data selection paradigm and the different proposed techniques.

We organized this chapter as follows. In the first place, Sections 3.2 and 3.3 provide an introduction to the adaptation problem in SMT. Sections 3.3.1 and **??** review the different adaptation methods in SMT and briefly list related works dealing with the adaptation problem. Finally, this thesis proposes different Data Selection techniques, applied to different tasks. For this reason, in this chapter we provide an

introduction to the data selection problem (Section 3.4, and present different state-of-the-art methods (Section 3.4.1 and 3.4.2) we use to compare our methods.

Table 3.1 shows the abbreviations introduced in the current chapter, in order to facilitate a better comprehension of the text.

Table 3.1. Abbreviations used in Chapter 3.

Abbreviation	Description
SMT	Statistical Machine Translation
DS	Data Selection
LM	Language Model
CE	Cross-Entropy method
PBSMT	Phrase-based Statistical Machine Translation
NMT	Neural Machine Translation

3.2 Adaptation

Domain Adaptation [99] is a field associated with machine learning and transfer learning. All machine learning methods work well only under a specific assumption: the training and test data are drawn from the same distribution. When the domain changes, most statistical models must be rebuilt for the new domain. In many real-world applications, it is expensive or impossible to obtain the new data needed to reconstruct the machine learning models for the new domain. Therefore, it is necessary to develop approaches that reduce the effort of adapting models to a new domain. Algorithms that tackle this problem are usually called domain adaptation algorithms. For instance, a classic example available in the literature is related to the common spam filtering problem, and consists in adapting a model from one user to a new one who receives significantly different emails [99, 100]. Domain adaptation involves two interrelated problems, aiming to learn a robust classifier in the source domain hoping that it will perform well in the related target domain by reducing the discrepancy between their distributions. There are two main categories for domain adaptation algorithms:

- semi-supervised domain adaptation: a small number of instances in the target domain are labeled.

- unsupervised domain adaptation: instances in the target domain data are completely unlabeled.

The last scenario is a topic of ongoing interest among researchers as it reflects what actually happens when a system trained under perfect conditions faces reality.

3.3 Adaptation in SMT

Domain adaptation is an active topic in the SMT community and receives considerable attention. The parallel corpora for building an SMT system can be obtained from different ways: websites, legal documents, proceedings from the European parliament and the United Nations, or even technical documents such as printer manuals, etc. The adaption problem appears when the corpus used to create or adjust the SMT models belongs to a different domain than the test domain to be translated.

In this chapter, we review previous works regarding the adaptation problem in machine translation. Adaptation techniques are classified in two different lines:

1. Off-line adaptation: which is applied on data available before deploying the SMT models.

2. On-line adaption: which is applied on data available while deploying the SMT models.

In this work, we will refer to the available pool of generic-domain sentences as out-of-domain corpus because we assume that it belongs to a different domain than the one to be translated. Similarly, we refer to the corpus belonging to a specific domain of the text to be translated as in-domain corpus.

3.3.1 OFF-LINE ADAPTATION

This section describes the research on off-line adaptation methods that are applied before deploying the SMT system. We divide this adaptation technique in two different categories, i.e. domain and topic. Domain and topic adaptations can be seen as complementary methods to cope with the variability between training and testing data in SMT. Domain adaptation typically assumes training data partitioned according to human-defined labels, while topic adaptation learns the corpus labels by data clustering [101].

3.3.1.1 Domain adaptation

Domain adaptation methods in SMT can be split into two broad categories [101, 102]:

1. Domain adaptation methods that tackle the problem at the corpus level, e.g., by weighting, selecting or joining the corpora.

2. Domain adaptation methods that have an influence at the model level by combining multiple translations or language models together, often in a weighted manner.

In this section, we will present an executive list of works in different SMT techniques.

Research on mixture models has considered both linear and log-linear mixtures [103, 104]. In [105], two basic settings are investigated: cross-domain adaptation, in which a small sample of parallel in-domain text is assumed, and dynamic adaptation where only the current input source text is considered. Adaptation relies on mixture models estimated on the training data through some unsupervised clustering method. Given the available adaptation data, mixture weights are re-estimated. A variation of this approach was proposed by [106]. In [107], mixture models are instead employed to adapt a word alignment model to in-domain parallel data.

Exploiting an in-domain monolingual corpus is also an effective approach to domain adaptation for SMT. [108–110] used monolingual corpora to generate a synthetic bilingual data through an SMT system.

Empirical results show that having an in-domain monolingual corpus could substantially improve translation quality, especially with in-domain monolingual data on the target side [111]. There are other ways of adapting translation models with monolingual corpora with different degrees of success [112–115].

Other methods can be performed at the target language model level [116,117] by selecting data using information retrieval techniques with the consequent adaptation to the language model. Language model adaptation has been deeply explored in the SMT community. [116] propose to build a query from a list of candidate translations for each source sentence. Such query is used to retrieve similar sentences from a very large training corpus. The retrieved sentences are used to build specific LMs which are then interpolated in translation time with a background LM estimated on all the data available.

Different publications compared different domain adaptation methods such as [118–120]. In [119], the authors compared different domain adaptation methods in the patent domain and observe small gains over the baseline. In [118] different adaptation techniques were applied on a phrase-based SMT system trained on the Europarl corpus, in order to adapt it to the news domain. In particular, a small portion of the in-domain corpus was exploited to adapt the Europarl language model and translation models by means of linear interpolation techniques. In [120] explored an empirical comparative of different domain adaptation method in the NMT paradigm.

In the next section we devote particular attention to Data Selection techniques.

3.4 Data selection

In the previous Section 3.3.1.1, we exposed that the size of the training corpus and the domain has an important impact in the final translation quality [121]. In these circumstances, specialised domain adaptation techniques are required to effectively utilise the available out-of-domain parallel data to improve in-domain translation quality. Data selection (DS) [101, 102, 122] is a method that uses out-of-domain corpora to complement the available, potentially sparse, in-domain

corpus. However, simply combining large amounts of supplementary out-of-domain data with small amounts of in-domain data might negatively affect translation quality by overwhelming the in-domain characteristics in the SMT system [104]. Therefore, relevant data selection is necessary, where only the best part of the out-of-domain corpus, the part that is similar to the specific domain at hand, supplements the in-domain training corpus.

A wide variety of selection methods have been used over the years. The main principle is to measure the similarity of sentences from the out-of-domain data with respect to the in-domain data, either regarding the development or the (source side of the) test set. Such similarity is often based on information theory metrics, as perplexity, applied to either side of the training data (source or target) or both. The selected out-of-domain sentences are eventually used to enhance an existing baseline SMT model in order to improve in-domain translation quality.

Techniques based on information retrieval have also been widely used for data selection. The information retrieval methods often use Term Frequency, Inverse Document Frequency (TF-IDF), which is a numerical statistic that intends to reflect how important a word is to a document in a collection or corpus. One focus for these methods is mixture modelling, wherein data is selected to build sub-models that are combined into one model that is in-domain. [123] use information retrieval techniques to select parallel training data that is most similar to a given test set. [124] use an information retrieval system to assign weights to each training sentence pair according to their similarity to the sentences in a given test set, prior to estimating the translation model.

In the last years, perplexity-based methods have become very common [125–130]. The first work was proposed by [125], where the sentences in an out-of-domain corpus are ranked by their perplexity score according to an in-domain language model, and only the top percentage with lowest perplexity scores are retained as training data. Similarly, [131] used the average perplexity derived from language model perplexities on both source and target sides of parallel datasets to select supplementary training data and combine them with smaller

in-domain translation models. This was accomplished using linear interpolation, extending the approach of [126] to use the difference of cross-entropy of supplementary sentence pairs on in-domain and out-of-domain corpora for data selection. [127] presented an adaptation approach based on bilingual cross-entropy difference and reported significant improvements over other similar models. These two techniques will be explained in detail in Section 3.4.1.

Other approaches are based in out of vocabulary words with respect to the training data [129, 132, 133], which were also reported to be successful triggers for supplementary data selection leading to improvements in both language and translation quality performance. In the context of adaptation, [134] saw the problem from a different perspective and proposed a quality estimation model.

Two different approaches are presented by [135]: one based on approximating the probability of an in-domain corpus and another one based on infrequent n-gram recovery. On the one hand, the former relies on preserving the probability distribution of the task domain by wisely selecting the bilingual pairs to be used, excluding sentences that distort the actual probability. On the other hand, the latest technique (the best-performing one) is based on the notion of infrequent n-gram and will be explained in detail in Section 3.4.2.

Finally, distributed representations of words have proliferated during the last years in the research community. Furthermore, [136] leveraged neural language models to perform DS reporting substantial gains over conventional DS using n-gram language models. Recently, convolutional neural networks have also been used in data selection [137, 138] obtaining positive results.

3.4.1 Cross-Entropy based methods

In Section 3.4, we mentioned previously existing methods for selecting the best part of data from an out-of-domain training corpus. The most established technique consists on ranking the sentences by their perplexity score according to a LM [125]. This method selects only those sentences with the lowest perplexity scores. The idea was that only sentences similar to the in-domain corpus would remain; reducing, at the same time, the perplexity of the corpus of mixed-domain

sentences compared to all the available corpus. The method proposed by [126] is a re-implementation of the perplexity-based method, with the principal difference being the use of the sentence cross-entropy rather than the perplexity, even though they are both monotonically related. The cross-entropy $H_C(\mathbf{x})$ of a given sentence $\mathbf{x} = \{x_1, \ldots, x_I\}$, according to a given language model p estimated on corpus C, is defined as follows:

$$H_C(\mathbf{x}) = -\sum_{i=1}^{|\mathbf{x}|} \frac{1}{|\mathbf{x}|} \log p(x_i \mid x_1, \ldots, x_{i-1}) \tag{3.1}$$

Then, the formulation proposed by [126] is: let D be an in-domain source corpus , and G be an out-of-domain source corpus from which we draw sentence $\mathbf{x}$. The cross-entropy difference score of $\mathbf{x}$ is then defined as:

$$c(\mathbf{x}) = H_D(\mathbf{x}) - H_G(\mathbf{x}) \tag{3.2}$$

Let $H_D(\mathbf{x})$ be the cross-entropy of a sentence $\mathbf{x}$ drawn from corpus G, according to a language model trained on D. Similarly, $H_G(\mathbf{x})$ be the cross-entropy of $\mathbf{x}$ according to a language model trained on G. Lower scores indicate more relevant sentences. Although the cross-entropy difference method is described as selecting sentences that are unlike the distribution of the out-of-domain corpus, it is more accurate to say it selects sentences which are closer to the in-domain distribution than to the generic distribution.

3.4.1.1 Bilingual cross-entropy method

In [127], the authors propose an extension of the cross-entropy method proposed in [126], that is able to deal with bilingual information. To this end, the sum of the cross-entropy difference over each side of the corpus, both source and target, is computed. Let (D and G) be an in-domain source corpus and an out-of-domain source corpus respectively, and (L and E) be the corresponding target corpora. Then, the bilingual cross-entropy difference is defined as:

$$c(\mathbf{x}, \mathbf{y}) = [H_D(\mathbf{x}) - H_G(\mathbf{x})] + [H_L(\mathbf{y}) - H_E(\mathbf{y})] \tag{3.3}$$

3.4.2 INFREQUENT NGRAMS RECOVERY

The main idea underlying the infrequent n-grams recovery strategy [135] consists in increasing the information of the in-domain corpus by selecting sentences from a out-of-domain corpus to maximise the coverage of n-grams wich appear in the test corpus. The n-grams that have never been seen or have been seen just a few times are called *infrequent n-grams*. For this, it is necessary to establish the infrequency threshold t required for a certain n-gram to be considered as infrequent, and also the order n of the n-grams (unigrams, bigrams, 3-grams etc.) that will be considered. The selected sentences will contain n-grams considered infrequent. With that we ensure that the training set will contain all n-grams from test set t times, as long as this is possible with the available out-of-domain corpus. Sentences in the out-of-domain corpus are sorted by their infrequency score $i(\mathbf{x})$ in order to select first the sentences which most improve the coverage of n-grams belonging to the in-domain corpus which might be considered infrequent.

Let X be the set of n-grams that appear in the sentences to be translated or source test corpus T, and m one of them; let be $R(m)$ the counts of m in a given source sentence $\mathbf{x}$ of the out-of-domain corpus, and $C(m)$ the counts of m in the source language in-domain corpus. Then, the infrequency score $i(\mathbf{x})$ is defined as:

$$i(\mathbf{x}) = \sum_{m \in X} \min(1, R(m)) \max(0, t - C(m)) \tag{3.4}$$

The sentences in the out-of-domain corpus are scored using Equation 3.4 as follows: the sentence $\mathbf{x}^*$ with the highest score $i(\mathbf{x}^*)$ is selected in each iteration. $\mathbf{x}^*$ is added to the in-domain corpus and is removed from the out-of-domain sentences. The counts of the n-grams $C(m)$ are updated with the counts $R(m)$ within $\mathbf{x}^*$ and therefore the scores of the out-of-domain corpus are updated. Note that t will determine the maximum amount of sentences that can be selected, since when all the n-grams within X reach the t frequency no more sentences will be extracted from the out-of-domain corpus.

3.5 Domain adaptation in NMT

As presented all over this chapter, the adaptation problem appears when the training data domain is different from the target domain. In PBSMT, an extensive number of methods for domain adaptation have been proposed (see Section 3.3.1.1). In the case of NMT, this task has been recently receiving an increasing interest [79].

For NMT, a fairly simple method is currently the most popular, called fine-tuning [79]. This method divides the training process into two steps. First, we train the NMT model on all out-of-domain available data until convergence. Then, we run a few more iterations of training on the in-domain corpus only and stop training when performance on the in-domain validation corpus starts decreasing. With this method, the final NMT system benefits from all the training data but is still adapted to the in-domain data [110, 120, 139–141].

Other methods used when dealing with the domain adaptation problem in NMT draw on the idea of ensemble decoding [142–144]. Training separate models for different data corpora (in-domain corpus and out-domain corpus), we may combine their models, just as we did for ensemble decoding. Other adaptation methods are based on the domain information or make use of the domain of the input sentence [145, 146] by adding a domain token to each training and test sentence. Chen *et.al.*, [147] report better results over this method of adaptation by encoding the given domain of each sentence as an additional input vector to the conditioning context of the word prediction layer.

Finally, [110] showed that parallel data is not strictly necessary for performing domain adaptation: the usage of synthetic data has positive effects on the NMT system. For obtaining the synthetic data they automatically translated a large monolingual corpus. This synthetic-based approach yielded better results than other methods aimed to exploit monolingual data [148, 149].

Given the important impact of the NMT models in state-of-the-art MT, we think that domain adaptation and more specifically DS, need special attention in NMT. For this reason, we dedicate a chapter of this thesis to this task (see Chapter 7). In that chapter, we explain in

detail the task and techniques proposed to solve the domain adaptation problem in NMT employing DS methods.

3.6 Summary

In this chapter, we have introduced two important concepts that constitute the basis of different works within this thesis. On the one hand, we presented the domain adaptation problem in SMT, as well as different approaches existing in the literature. On the other hand, we paid particular attention to the data selection approach, a paradigm that encompasses the various adaptation techniques proposed in this thesis.

4 Corpus selection for SMT training

* * *

"Wake up, Alice dear!" said her sister; "Why, what a long sleep you've had!"

—LEWIS CARROLL
ALICE'S ADVENTURES IN WONDERLAND

"¡Despierte, Alicia estimada! dicho su hermana; "¡ Porqué, qué un sueño largo usted ha tenido!"

—SYSTRANET TRANSLATOR
LAS AVENTURAS DE ALICIA EN EL PAÍS DE LAS MARAVILLAS

* * *

4.1 Introduction

As introduced in Section 3.4, DS implies selecting (for training) the best subset of sentence pairs from an available pool so that the translation quality achieved in the target domain is improved.

The current chapter tackles DS by taking advantage of neural networks. The ultimate goal is to obtain corpus subsets that minimise the bilingual corpus training size, while improving translation quality. We have named the first DS technique proposed Continuous Vector-Space Representation of Sentences for Data Selection (CRSDS), with the aim of selecting the best subset of sentences using a vector space representation of sentences or words. This method represents the most recent work performed on distributed representations of words or sentences [55, 64] with the goal of obtaining a vectorial represen-

tation where the syntactic and semantic relationships between words are preserved.

The second DS technique proposed is a Neural Network Classifier of Sentences for Data Selection (NNCDS). The idea is to view the DS problem as a classification task with the goal of classifying the out-of-domain sentences into in-domain, or purely out-of-domain. The rest of the chapter is organized as follows: Sections 4.2 and 4.3 present our different DS methods used to obtain the best subset corpus. Experiments and discussions are presented in Section 4.4 and conclusions drawn from results are presented in Section 4.5.

Table 4.1 shows the abbreviations introduced in the current chapter, in order to facilitate a better comprehension of the text.

Table 4.1. Abbreviations used in Chapter 4.

Abbreviation	Description
SMT	Statistical Machine Translation
DS	Data selection
CVR	continuous vector-space representation
CRSDS	Continuous Vector-Space Representation of Sentences for DS
NNCDS	Neural Network Classifier of Sentences for DS
`Mean-vec`	sentence embedding method
`Document-vec`	sentence embedding method
CNN	Convolutional neural networks
BLSTM	Bidirectional LSTM networks
CE	Cross-Entropy method
RNN	Recurrent Neural Networks

4.2 CRSDS technique

In this section, we introduce the Continuous Vector-Space Representation of Sentences for Data Selection (CRSDS) technique. To define our strategy, the following details are required:

1. Similarity corpus (Section 4.2.1)
2. Sentence embedding (using step 1) (Section 4.2.2)
3. A selection algorithm (using step 2) (Section 4.2.3)

4.2.1 SIMILARITY CORPUS

With the purpose of simplifying notation, we start by defining the notion of similarity corpus (S).

The core idea of every DS method is to select a subset of the out-of-domain data that is considered to be the most relevant for translating a given set of data, named in this work *similarity corpus* S. Ideally, S will be the text to be translated (T), and the DS method will ensure that the resulting subset of the training data is the best possible subset for translating T [135]. Nevertheless, in scenarios where a system is set for on-the-fly translation, such data T is not available in advance. Thus, it is often the case where an in-domain set D (considered to be very similar, or at least belonging to the same domain as T) is used instead [124,127]. We will define our CRSDS technique independently of whether D or T is used, our data selection method will be defined in terms of S and the experimental results will instantiate S to either D or T. Note that T lacks of an important piece of information present in D: the target side of the bilingual data. In contrast, T contains the true data to be translated, albeit obviously, without the output sentence.

4.2.2 SENTENCES EMBEDDING METHODS

Section 1.4.3 presents different methods for CVR of sentences. For this technique, two methods are used: `Mean-vec` and `Document-vec`.

4.2.3 CRSDS TECHNIQUE

In this section, we will describe the CRSDS method which leverages the sentence embedding ($F_{\mathbf{x}}$), described above. Considering the objective of DS is to increase the informativeness of the in-domain training corpus, it seems important to choose out-of-domain sentences that provide information considered relevant with respect to the similarity corpus S.

Algorithm 1 shows this procedure. Here, G is the out-domain-corpus, $\mathbf{x}$ is an out-of-domain sentence ($\mathbf{x} \in G$), $F_{\mathbf{x}}$ is the CVR of $\mathbf{x}$ obtained with some methods in (Section 1.4.3), and $|G|$ is the number of sentences in G. Then, our objective is to select the most suitable data from G for translating data belonging to the similarity corpus S.

This selected data is defined as `Selected-corpus`. For this purpose, we define F_s as the CVR of a sentence $\mathbf{s} \in S$.

Data: $F_{\mathbf{x}}$, $\mathbf{x} \in G$; and $F_{\mathbf{s}}$, $\mathbf{s} \in S$; threshold τ
Result: `Selected-corpus`
forall *sentence* **s** *in* S **do**
  forall *sentence* **x** *in* G **do**
    if $sim_i((F_{\mathbf{s}}, F_{\mathbf{x}}), \tau)$ **then**
      add **x** to `Selected-corpus`
      remove **x** to G
    end
  end
end

Algorithm 1: Pseudo-code for CRSDS technique (Section 4.2.3)

Algorithm 1 introduces $sim_i(\cdot, \cdot)$, which will be defined in Section 4.2.3.1.

4.2.3.1 *Similarity functions*

The most simple approach will be to implement a mechanism by which a sentence **x** would only be selected if its similarity score is: $cos(F_{\mathbf{s}}, F_{\mathbf{x}}) \geq \tau$, where τ is certain threshold established empirically, i.e:

$$sim_0((F_{\mathbf{s}}, F_{\mathbf{x}}), \tau) = \begin{cases} cos(F_{\mathbf{s}}, F_{\mathbf{x}}) & \text{if } cos(F_{\mathbf{s}}, F_{\mathbf{x}}) \geq \tau \end{cases} \tag{4.1}$$

The function $cos(\cdot, \cdot)$, represents the cosine similarity between two different sentence vectors:

$$cos(F_{\mathbf{s}}, F_{\mathbf{x}}) = \frac{F_{\mathbf{s}} \cdot F_{\mathbf{x}}}{\|F_{\mathbf{s}}\| \cdot \|F_{\mathbf{x}}\|} \tag{4.2}$$

Note that it is possible to use any other similarity metric. Here, the purpose of similarity function $sim(\cdot, \cdot)$ is to allow a projection from the original similarity metric, so as to allow higher flexibility. The *best* value for $cos(\cdot, \cdot)$ is 1 and the *worst* value for $cos(\cdot, \cdot)$ is 0.

Nevertheless, this approach proved not to be empirically useful: certain, very specific, sentences in S yield much higher similarity scores,

dominating the ranking when establishing τ and leading to other sentences in S not getting the chance to promote any sentences in G at all, i.e., a small number of sentences in S account for the wide majority of sentences selected. This is problematic, since the final set selected in such case is only suitable for translating a very small subset of S.

Hence, we developed three different similarity functions $sim_i(\cdot), i \in \{1,2,3\}$, for the metric $cos(F_{\mathbf{s}}, F_{\mathbf{x}})$ with the purpose of solving this issue. Let us first define $G_{s,\tau} = \{\mathbf{x} \mid \forall \mathbf{x} \in G : cos(F_{\mathbf{s}}, F_{\mathbf{x}}) > \tau\}$. Then, the similarity functions used are defined as follows:

sim_1 The purpose of this approach is to limit the amount of sentences $\mathbf{x} \in G$ that can be promoted by a certain sentence $\mathbf{s} \in S$. Let μ be the empirical average of $|G_{\mathbf{s},\tau}|$, i.e., $\mu = \sum_{\mathbf{s}\in S} |G_{\mathbf{s},\tau}|/|S|$, and σ the corresponding standard deviation of $|G_{\mathbf{s},\tau}|$. Since $cos(F_{\mathbf{s}}, F_{\mathbf{x}})$ establishes a natural ordering in G for each $\mathbf{s} \in S$, let us define $G'_{\mathbf{s},\tau}$ as the set of sentences with highest $cos(F_{\mathbf{s}}, F_{\mathbf{x}})$ value, such that $|G'_{\mathbf{s},\tau}| \leq \mu + 2\sigma$. Then, we define sim_1 as follows:

$$sim_1((F_{\mathbf{s}}, F_{\mathbf{x}}), \tau) = \begin{cases} cos(F_{\mathbf{s}}, F_{\mathbf{x}}) & \text{if } \mathbf{x} \in G'_{\mathbf{s},\tau} \\ 0 & \text{if } \mathbf{x} \notin G'_{\mathbf{s},\tau} \end{cases} \tag{4.3}$$

sim_2 In this case, the purpose is to promote those sentences in G that are the most similar to the whole similarity corpus S. We implemented this intuitive concept as the arithmetic mean of $cos(\cdot,\cdot)$ for all sentences $\mathbf{s} \in S$, i.e.:

$$sim_2((F_{\mathbf{s}}, F_{\mathbf{x}}), \tau) = \frac{\sum_{\mathbf{s}\in S} cos(F_{\mathbf{s}}, F_{\mathbf{x}})}{|S|} \tag{4.4}$$

sim_3 This proposal is dramatically different from the previous ones, $cos(F_{\mathbf{s}}, F_{\mathbf{x}})$ is not employed as such. Instead, we computed a CVR of the whole corpus S, $F_{\mathbf{S}}$, assuming that S is the concatenation of all its sentences, and the threshold selection (line 4 of Algorithm 1) is applied on such score:

$$sim_3((F_{\mathbf{s}}, F_{\mathbf{x}}),\tau) = cos(F_{\mathbf{S}}, F_{\mathbf{x}}) \tag{4.5}$$

Notation has been slightly abused since S is not present in the parameter list of sim_3; it has been omitted for clarity.

4.2.4 BILINGUAL-CRSDS TECHNIQUE

In this section, we extend the CRSDS technique presented in Section 4.2.3 for making use of bilingual data, called Bilingual-CRSDS. Here, the purpose is to tackle directly the bilingual nature of the DS problem within an SMT setting by including both sides of the corpus (source and target sentences). Before describing this method, it is important to emphasize that the similarity corpus S includes, in this case, both the source and target in-domain corpus.

Algorithm 1 is modified as follows (Algorithm 2). Here, $\mathbf{x}_G$ is an source out-of-domain sentence ($\mathbf{x}_G \in G_x$), $F_{\mathbf{x}_G}$ is the CVR of $\mathbf{x}_G$ and $\mathbf{y}_G$ is an target out-of-domain sentence ($\mathbf{y}_G \in G_y$). Similarly as done for F_S , we define F_{S_x} as the sentence embedding of S_x, i.e., the CVR of the concatenation of all source in-domain data and F_{S_y} as the CVR of S_y, i.e., the CVR of the concatenation of all target in-domain data.

Input: $F_{\mathbf{x}_G}$, $\mathbf{x}_G \in G_x$, $F_{\mathbf{y}_G}$ $\mathbf{y}_G \in G_y$, and $F_{\mathbf{s}_x}$, $\mathbf{s}_\mathbf{x} \in S_x$; $F_{\mathbf{s}_y}$, $\mathbf{s}_y \in S_y$, threshold τ
Output: `Selected-corpus`

```
forall sentence x_G in G_x and y_G in G_y do
    if [cos(F_{s_x}, F_{x_G})] + [cos(F_{s_y}, F_{y_G})] ≥ τ then
        add x_G, y_G to Selected-corpus
        remove x_G, y_G to G_x, G_y
    end
end
```

Algorithm 2: Pseudo-code for Bilingual-CRSDS technique (Section 4.2.4)

4.3 NNCDS technique

In this section, we introduce a new DS technique called Neural Network Classifier for Data Selection (NNCDS). Here, we describe our NNCDS for SMT. To define the strategy, the following steps are required:

1. Neural network architecture (Section 4.3.1)
2. Semi-supervised algorithm (Section 4.3.2)

In this section, we tackle the DS problem as a classification task. Let us consider a classifier model M that assigns a probability $p_M(\mathbf{x})$ to a given sentence $\mathbf{x}$, depending on whether $\mathbf{x}$ belongs to the in-domain corpus D or not. In this case, to obtain the `Selected-corpus`, one could just apply the classifier M to each sentence from the out-of-domain pool G and select the most probable ones.

4.3.1 NEURAL NETWORK ARCHITECTURE

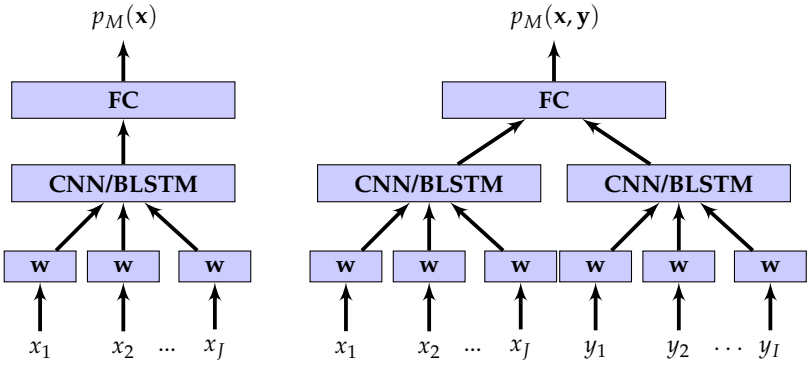
We propose to use a neural classifier for DS task, exploring two different neural networks (Convolutional Neural Network and Recurrent Neural Network) as sentence encoders. In Figure 4.1, we show the general architecture of our neural classifier. In this model, the input sentence $\mathbf{x}$ is fed to our system following a one-hot codification scheme and is projected to a continuous space using a word-embedding matrix. Next, the sequence of word embeddings is processed either by a Convolutional neural network (CNN) or a Bidirectional Long Short-Term Memory network (BLSTM). Next, we stack one or more fully-connected (FC) layers. Finally, we can apply a softmax function, if we wish to obtain normalized probabilities. All elements can be jointly trained by maximum likelihood.

This reasoning can be extended in order to apply it to a bilingual corpus, as shown in Figure 4.1. Therefore, if we have the source sentence $\mathbf{x}$ and its corresponding translation $\mathbf{y}$, we can model the probability $p_M(\mathbf{x}, \mathbf{y})$. To accomplish this, we used two networks, one for the source language and another one for the target language. Then, we concatenated their outputs and apply FC layers, as in the previous case, computing a unique score for each bilingual pair.

Convolutional neural networks

CNNs have proven their representation capacity, not only in computer vision tasks [150], but also depicting text [63,71,77].

In this work, we used the non-static CNN proposed by [63]. This CNN consists in the application of a set of filters to windows of different lengths. These filters apply a non-linear function (e.g. ReLU). Next, a max-pooling operation is applied to the set of convolutional



(a) Monolingual Model **(b)** Bilingual Model

Figure 4.1. General architecture of the proposed NNCDS technique. The monolingual model is shown at the left while the bilingual model is shown at the right.

filters. As result, the CNN obtains a feature vector representing the input sentence $F_{\mathbf{x}}$.

Recurrent neural networks (RNN)

In RNNs, connections form a directed cycle. This fact allows the network to keep an internal state and be effective sequence modellers. Moreover, bidirectional networks [151] have two independent recurrent layers, one processing the input sequence in a forward manner and the other one processing it in a backward manner. Therefore, they allow to exploit the full context at each time-step. Gated units, such as Long Short-Term Memory (LSTM) [152, 153], mitigate the vanishing gradient problem and hence, they are able to properly model long sequences. The vanishing gradient problem is a hindrance found when training certain neural networks with gradient based methods (e.g back-propagation). In particular, this problem makes it really hard to learn and tune the parameters of the earlier layers in the network. It becomes worse as the number of layers in the architecture increase. In this work, we used BLSTM networks [154] for encoding a sentence by concatenating the last hidden state of the forward and backward LSTM layers. In this way, a compact CVR of the sentence is provided, which accounts for relationships in both time directions.

4.3.2 SEMI-SUPERVISED SELECTION

Properly training these neural classifiers may be a challenging task, since the in-domain data is scarce. Hence, for training them, we follow a semi-supervised iterative protocol [155].

Input: P_0 (positive samples), N_0 (negative samples), G_0 (out-of-domain corpus), l (selection size), r (training granularity)
Output: P_i (selection of size l)

begin
  $i = 0$
  while $|P_i| \leq l$ **do**
    $M_i \leftarrow$ `Train model on` $\{P_i \cup N_i\}$
    $S_i \leftarrow$ `Classify` G_i `with` M_i
    $P_{i+1} \leftarrow \{P_i \cup$ `get_top`$(S_i, r)\}$
    $N_{i+1} \leftarrow \{N_i \cup$ `get_bottom`$(S_i, r)\}$
    $G_{i+1} \leftarrow \{G_i -$ `get_top`$(S_i, r) -$ `get_bottom`$(S_i, r)\}$
    $i++$
  end
  return P_i
end

Algorithm 3: Semi-supervised selection for NNCDS. The functions `get_top` and `get_bottom` select the top-r and the bottom-r scoring sentences from a scored set. The algorithm returns a selection consisting of l sentences.

Algorithm 3 shows this semi-supervised training procedure. Since data selection is a binary classification problem, we need a set of positive and negative training samples. The algorithm starts from an initial set of positive samples P_0 and a set of negative samples N_0. A major step is how we select our initial P_0 and N_0. We set our in-domain corpus D as P_0. We randomly extract $\mid D \mid$ sentences from G for constructing N_0. The initial out-of-domain corpus G_0 is defined as $\{G - N_0\}$. At each iteration $i \geq 0$, we train a model with the current sets of data (P_i, N_i), this step is represented by `Train model`. Next, with the `Classify` function, we classify all sentences belonging to the out-of-domain pool (G_i). We extract a number r of top-scoring sen-

tences and include them into the set of positive samples, producing a new set P_{i+1}. Analogously, the r bottom-scoring sentences are included into a new negative samples set N_{i+1}. These processes are performed in functions `get_top` and `get_bottom` respectively. Hence, at each iteration, we remove $2r$ samples from the out-of-domain set, producing the pool G_{i+1}. Then, a new iteration starts. This is repeated until the selection P_i reaches the desired size (l).

4.4 Experiments

In this section, we describe the experimental framework used to assess the performance of the data selection method described in Section 3.4. Then, we show the results of CRSDS strategy, followed by a comparative with two state-of-the-art data selection methods (cross-entropy method and infrequent ngrams recovery).

In order to compare different DS methods, we explored the effect of varying empirically the selection constraint (e.g., the maximum number of selected sentences in Section 3.4.1, the threshold τ in Section 4.2.3 or neural network architecture in Section 4.3.1). These preliminary experiments were conducted on different domains and language pairs. By doing so, we obtained different subsets of the selected out-of-domain corpus. Then, a SMT system is trained on each selected subset and tested on the development corpus. This provides several comparison points between the DS methods. In this setting, the different selection methods are compared based on how many sentences are required in order to reach the evaluation metric score.

4.4.1 Experimental setup

We evaluated empirically the DS methods described in Section 4.2, Section 4.3, Section 3.4.1 and Section 3.4.2. As explained before, SMT systems need large corpora for training the underlying statistical models. Two corpora are dealt within the DS task: an out-of-domain corpus and an in-domain corpus in Section 3.4. DS selects only a portion of the out-of-domain corpus, and leverages that subset together with the in-domain data to train a hopefully improved SMT system.

The DS methods were evaluated in two different domains (Medical domain and IT domain). We conducted each domain experiment using different language pairs with the purpose of evaluating whether the conclusions drawn from one single language pair hold in further scenarios.

- **Medical domain**: As in-domain data, we used the EMEA corpus. The test and development corpora are the partitions established in the 2014 WMT. Corpora details are given in Section 2.3. In this domain, we focused on English, French, and German languages; across four different language directions: EN→FR, FR→EN, EN→DE and DE→EN.

- **IT domain**: DS techniques were evaluated on the IT domain. Details are shown in Section 2.3. The languages assessed, in this case, were English, Spanish and Czech; across different language directions EN→ES, ES→EN, EN→CS and CS→EN.

Finally, as out-of-domain corpus we used the Europarl corpus readily available in the literature, details are given in Section 2.3.

In the CRSDS technique, the word embeddings were trained by word2vec toolkit (Section 2.4), using the Skip-Gram model (details in Section 1.4.2).

In the NNCDS method, all neural models were initialized using word embeddings matrices from the word2vec toolkit and trained on Google News dataset (English) and on Wikipedia (others languages). Word embeddings matrices were fine-tuned during the semi-supervised selection protocol. The size of the words embeddings was, $size = 200$. To train the CNN classifier, we used Adadelta algorithm [156] with its default parameters. The BLSTM network [154] was trained with the Adam algorithm [157], with a learning rate of 10^{-4}. During training, we applied Gaussian noise to the weights ($\sigma = 0.01$). All neural models were implemented using the Theano [158] and Keras libraries. Another important parameter for this method is the number of sentences selected at each iteration (r), which was chosen trading off speed and granularity ($r = 50k$).

Two different baselines were trained for each domain and compared to the systems obtained by DS methods. The first baseline was obtained by training the SMT system only with in-domain training data (EMEA or IT training corpus), obtaining the bsln-emea and bsln-it baselines, respectively. The second baseline was obtained by training the SMT system with a concatenation of either out-of-domain corpus (Europarl training corpus) and the in-domain training data (EMEA or IT training corpus) getting the bsln-emea-euro and bsln-it-euro. We also include results from a random sentence selection without replacement.

Finally, SMT translation output will be evaluated using BLEU [80], METEOR [82] and TER [83]. More details about these metrics are provided in Section 2.2.

4.4.2 CRSDS EXPERIMENTAL RESULTS

4.4.2.1 Different vector size and number of words

As a first step for empirical evaluation of the CRSDS technique, we analysed the effect of the different parameters that need to be adjusted to calculate the word and sentence embeddings. In this case, vector dimension *size* and n_c is the minimum number of times a given word needs to appear in the training data for its corresponding vector to be built. Table 4.2 shows the best results obtained with different vector sizes and n_c, all results were obtained using the development corpus. Experimental results are shown for Medical domain taking advantage of the similarity function sim_0 (see Section 4.2.3.1) with two different sentence embedding methods and for two language pairs. The similarity corpus S considered was the in-domain data D. A few conclusions can be drawn from the table:

- Translation quality could be affected when the value of n_c increases. This shows the necessity to use the highest vocabulary to obtain the better sentence representation. The value of $n_c = 1$ will be fixed for next experiments reported in this thesis.
- Note that translation quality is quite similar even though the vector size changes in $size = \{100, 500, 1000\}$. For this reason, the vector size was set to $size = 200$, in the next experiments.

Table 4.2. Translation results using CRDS method, variating the vector size and number of words. `Mean` and `Doc` are the two different CVR methods, *size* denotes the vector dimension size, n_c denotes the minimum number of times a given word needs to appear in the data. $|S|$ stands for the number of sentences, which are given in terms of the in-domain corpus size and (+) is the number of sentences selected.

			EN-FR			EN-DE		
Method	n_c	*size*	BLEU	TER	METEOR	BLEU	TER	METEOR
`Mean-`sim_0	1	50	29.8	50.8	52.2	16.0	39.4	64.1
		200	30.5	51.4	51.2	16.5	39.7	64.3
		500	30.6	51.3	51.1	16.4	39.8	64.5
		1000	30.6	51.4	51.0	16.6	39.8	64.4
	5	50	28.5	49.8	52.8	15.8	37.9	64.8
		200	29.6	49.2	52.3	16.0	38.2	65.8
		500	29.5	50.0	51.9	16.1	38.4	65.9
		1000	29.6	50.2	50.8	16.0	38.1	66.1
	10	50	27.7	49.2	53.3	15.1	37.2	66.1
		200	28.2	49.5	53.0	16.1	37.9	65.7
		500	28.5	49.3	52.7	16.0	38.0	65.1
		1000	28.6	49.8	52.5	16.1	37.9	65.2
`Doc-`sim_0	1	50	30.1	50.7	51.8	15.8	38.0	65.8
		200	30.9	51.8	50.8	16.4	39.1	65.3
		500	30.9	52.0	51.1	16.3	39.2	65.2
		1000	31.1	52.1	50.9	16.4	39.0	65.7
	5	50	28.9	49.8	52.8	15.7	37.8	65.0
		200	29.8	49.2	52.3	15.9	38.2	65.8
		500	29.7	50.3	51.2	16.1	38.4	65.9
		1000	29.5	50.2	50.8	16.0	38.1	66.1
	10	50	27.8	49.2	53.3	15.3	37.5	66.0
		200	28.3	49.5	53.0	15.9	37.9	65.8
		500	28.4	49.4	52.9	16.0	37.9	65.3
		1000	28.3	49.5	52.8	16.0	37.9	65.2

4.4.2.2 Vector representation and similarity function comparative

In this CRSDS analysis step, we studied the performance of two different CVR of sentences (`Mean-vec` and `Document-vec`). These two methods have a great impact on the vectors obtained, and are bound to have an important impact on the data selection technique, and finally in the translation quality.

Table 4.3 shows the best results for development corpus obtained with the different CVR methods. We only show the experimental results in the Medical domain using the three different functions sim_i (see Section 4.2.3.1) and two language pairs. The values show the best results for each strategy in terms of BLEU, and comparing the size of selected corpora. Note that translation quality remains quite similar, since the purpose of these experiments is to analyse the extent to which the different DS strategies are able to reduce the amount of training data required, without any significant loss in translation quality. In this case, the similarity corpus S considered was the in-domain data D.

Table 4.3. Translation results using CRDS method, in different configurations. `Mean` and `Doc` are the two different CVR methods, sim_i denotes the three different similarity functions, $|S|$ for number of sentences, which are given in terms of the in-domain corpus size, and (+) the number of sentences selected.

	EN-FR				EN-DE			
Method	$\|S\|$	BLEU	METEOR	TER	$\|S\|$	BLEU	METEOR	TER
bsln-emea	1.0M	27.5	48.1	54.5	1.0M	14.8	38.2	65.6
bsln-emea-euro	1.0M+1.5M	30.5	52.8	50.3	1.0M+1.5M	16.2	39.9	63.7
`Mean`-sim_0	1.0M+500k	30.5	51.4	51.2	1.0M+500k	16.1	38.8	64.7
`Mean`-sim_1	1.0M+347k	30.2	51.4	51.0	**1.0**M+**357k**	**16.5**	**39.7**	**64.3**
`Mean`-sim_2	1.0M+472k	30.9	51.9	50.9	1.0M+347k	16.2	39.4	63.1
`Mean`-sim_3	**1.0**M+**500k**	**31.0**	**52.3**	**50.3**	1.0M+400k	16.3	39.2	64.7
`Doc`-sim_0	1.0M+500k	30.9	51.8	50.8	1.0M+500k	16.2	39.2	64.8
`Doc`-sim_1	1.0M+384k	31.4	52.3	50.2	1.0M+440k	16.4	39.1	65.3
`Doc`-sim_2	1.0M+380k	31.1	52.2	50.1	1.0M+410k	16.3	39.7	64.9
`Doc`-sim_3	**1.0**M+**485k**	**31.6**	**52.8**	**49.8**	**1.0**M+**350k**	**16.4**	**39.9**	**64.2**

Several conclusions can be drawn:

- Translation quality using DS significantly improves over baseline (bsln-emea) translation quality.
- In EN-FR and EN-DE, translation quality using DS improves over (bsln-all), but using a significantly less amount of data (3% and 23%, respectively). In the case of DE-EN, translation quality results are similar, but using only 27% of the data. Hence, we can safely state that our DS strategy is always able to deliver

similar quality values to that obtained using all the data, but only with a rough quarter of the data.

- `Document-vec` yields slightly better translation quality than the `Mean-vec` method. Although differences are not statistically significant, this could mean that `Document-vec` entails a better estimation of the sentence CVR.

- Lastly, sim_1, sim_2 or sim_3 seem to perform similarly. However, sim_0 does require significantly more sentences to reach comparable translation quality. sim_3 should be preferred: it is the cheapest in computational terms because it only requires one comparison with each $\mathbf{s} \in S$.

4.4.3 NNCDS EXPERIMENTAL RESULTS

4.4.3.1 Neural networks comparative

In this NNCDS analysis step, we studied the performance of the two different neural network architectures (CNN and BLSTM). These two methods have a great impact on the neural classifier, and consequently in the translation quality. Table 4.4 shows the best results obtained for the Medical domain with two language pairs.

- The values show the best results in terms of BLEU and comparing the size of selected corpora. From the table, we can infer that both translation quality and selected corpus size are very similar, so we can not conclude which architecture is better for this task. Therefore, we decided to make all the experiments with both architectures.

4.4.4 COMPARATIVE DS METHOD USING THE IN-DOMAIN CORPUS

Once the effect of the different parameters in our DS methods (CRSDS and NNCDS) was analysed, we now pursue to compare our DS methods with the state-of-the-art Cross-Entropy method. In these methods, the selection process only considers the in-domain corpus. For this reason, the similarity corpus (S) defined by CRSDS method uses the in-domain corpus (i.e., $S = D$).

Table 4.4. Translation results using NNCDS method in different configurations. CNNs and BLSTM are the two different neural network architectures. $|S|$ stands for the number of sentences, which are given in terms of the in-domain corpus size, and (+) is the number of sentences selected.

	EN-FR				EN-DE			
Method	$\|S\|$	BLEU	METEOR	TER	$\|S\|$	BLEU	METEOR	TER
bsln-emea	1.0M	27.5	48.1	54.5	1.0M	14.8	38.2	65.6
bsln-emea-euro	1.0M+1.5M	30.5	52.8	50.3	1.0M+1.5M	16.2	39.9	63.7
CNNs	1.0M+450k	31.7	53.2	49.6	1.0M+350k	16.3	40.0	64.5
BLSTM	1.0M+300k	31.8	53.1	49.4	1.0M+400k	16.6	40.2	64.3

Besides, we divided this section in two parts, monolingual and bilingual:

- Monolingual DS methods comparative: The DS method only uses the source part of the corpus to select data.
- Bilingual DS methods comparative: DS method is able to use all the data available (source and target).

4.4.4.1 Monolingual DS method comparative

Medical domain results

Results in Figure 4.2 show the effect of adding sentences to the in-domain corpus. For the CRSDS method, we tested both CVR methods (`Mean-vec` and `Document-vec`, combined with sim_3). For the NNCDS method, we tested both network architectures (CNNs and BLSTM). Also, all results shown in this figure were obtained using the development corpus. Several conclusions can be drawn:

- All DS methods are mostly able to improve the random selection, especially when low amounts of data are added. Those cases where random selection was used yielded better results, although differences are not significant. This is reasonable since all DS methods including random selection will eventually converge to the same point: adding all the data available. Even

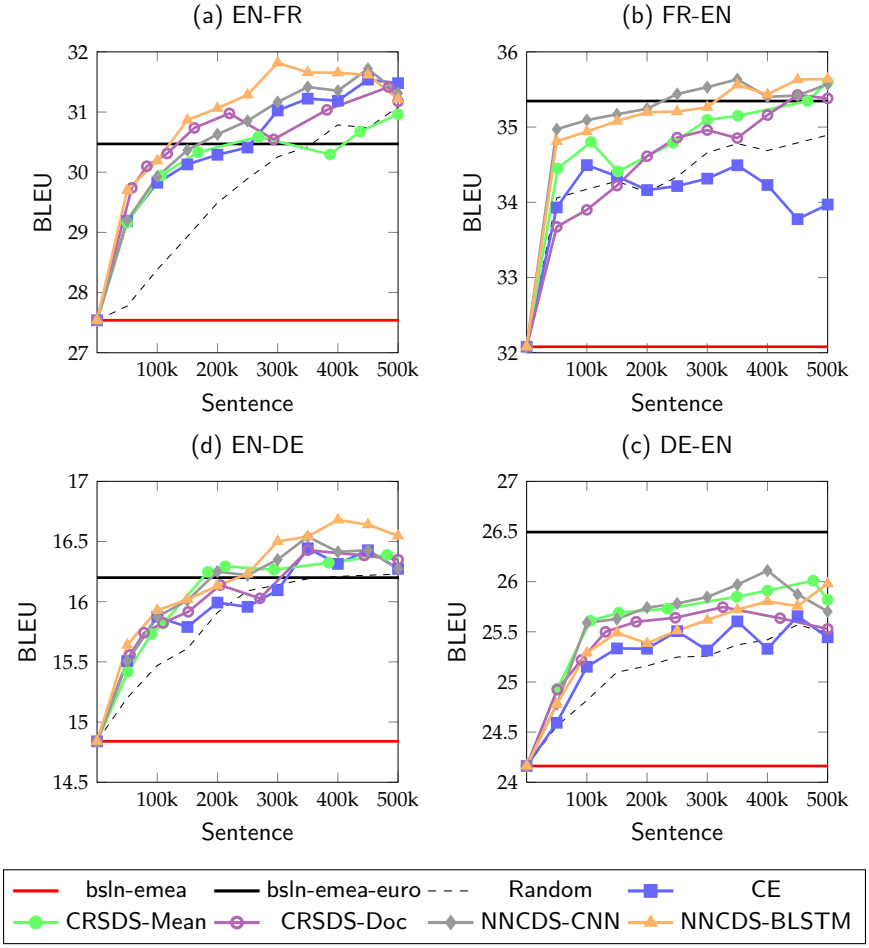


Figure 4.2. Graphical representation of the impact caused on BLEU metric by the addition of sentences to Medical domain using monolingual CRSDS, NNCDS, CE, and random selection. Horizontal lines represent bsln-emea and bsln-emea-euro.

though these results should be expected, previous works [159, 160] revealed that beating random selection was very hard.

- Results obtained with CRSDS method are slightly better (or similar) than the ones obtained with CE.

- In some cases, the performance obtained with NNCDS method is better than the one obtained with CE and our CRSDS allowing to reduce the number of sentences significantly.

Table 4.5 shows the translation results obtained for Medical domain test corpus. In this table, we can see that all the DS methods can achieve better translation results than the bsln-emea, across different language pairs. Our DS technique provides similar results to those including the full out-of-domain corpus (bsln-emea-euro) in language pairs EN-FR, FR-EN, and EN-DE, using less than $[38\% - 20\%]$ of the out-of-domain corpus. In the DE-EN pair our DS strategy does improve the results over including the full out-of-domain corpus, but results are very similar using less than 32% of the out-of-domain corpus.

Information Technology domain results

In this section, we show the results for the IT domain. Figure 4.3 shows the effect of adding sentences to the IT in-domain corpus. Experiments were made across four language pairs. In the CRSDS technique, we tested both CVR methods (`Mean-vec` and `Document-vec` combined with sim_3). For NNCDS, we tested both networks architectures (CNNs and BLSTM). All results shown in this figure were obtained using the development corpus. Several conclusions can be drawn:

- EN-ES, ES-EN and EN-CS, translation quality using DS method improves over bsl-it-euro at the beginning. In the case of CS-EN, translation quality results are similar to the ones obtained using bsl-it-euro but the size of the out-of-domain data used to train the system decreases.

- The case of EN-ES and ES-EN is very interesting. Experimental results show that using all the available data do not increase the translation quality (in terms of BLEU; bsl-it obtains better results than bsl-it-euro). The data selection method is able to obtain better results than bsl-it-euro.

Table 4.5. Best translation results for monolingual DS methods in test corpus of Medical domain. Columns denote, from left to right: Language pairs, DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected and BLEU, METEOR and TER are the evaluation metrics. CE stands for Cross-Entropy method, CRSDS for Continuous Vector-Space Representation of Sentences for Data Selection method and NNCDS for Neural Network Classifier for Data Selection method.

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**
EN-FR	bsln-emea	1.0M	28.6 ± 0.1	51.6 ± 0.1	52.7 ± 0.1
	bsln-emea-euro	1.0M+1.5M	29.4 ± 0.1	55.0 ± 0.1	50.2 ± 0.1
	Random	1.0M+500k	29.4 ± 0.3	54.9 ± 0.2	50.4 ± 0.1
	CE	1.0M+450k	29.8 ± 0.1	55.1 ± 0.1	50.3 ± 0.1
	CRSDS	1.0M+485k	29.7 ± 0.2	55.0 ± 0.2	50.3 ± 0.2
	NNCDS	**1.0**M+**300**k	$\mathbf{29.9 \pm 0.3}$	$\mathbf{55.2 \pm 0.2}$	$\mathbf{50.1 \pm 0.1}$
FR-EN	bsln-emea	1.0M	29.9 ± 0.2	35.4 ± 0.1	48.1 ± 0.2
	bsln-emea-euro	1.0M+1.5M	32.4 ± 0.1	37.6 ± 0.1	45.5 ± 0.1
	Random	1.0M+500k	32.3 ± 0.3	37.4 ± 0.1	45.5 ± 0.2
	CE	1.0M+500k	31.7 ± 0.1	37.0 ± 0.1	45.9 ± 0.3
	CRSDS	1.0M+500k	32.6 ± 0.2	37.7 ± 0.1	45.4 ± 0.2
	NNCDS	**1.0**M+**350**k	$\mathbf{32.3 \pm 0.2}$	$\mathbf{37.5 \pm 0.2}$	$\mathbf{45.5 \pm 0.3}$
EN-DE	bsln-emea	1.0M	15.4 ± 0.3	38.4 ± 0.2	65.4 ± 0.1
	bsln-emea-euro	1.0M+1.5M	16.6 ± 0.2	40.4 ± 0.2	64.4 ± 0.4
	Random	1.0M+500k	16.6 ± 0.1	40.5 ± 0.2	64.5 ± 0.3
	CE	1.0M+500k	16.8 ± 0.2	40.5 ± 0.1	64.4 ± 0.2
	CRSDS	1.0M+350k	16.7 ± 0.2	40.5 ± 0.2	64.4 ± 0.3
	NNCDS	**1.0**M+**400**k	$\mathbf{17.1 \pm 0.2}$	$\mathbf{40.8 \pm 0.2}$	$\mathbf{64.1 \pm 0.3}$
DE-EN	bsln-emea	1.0M	23.7 ± 0.2	29.9 ± 0.1	57.1 ± 0.6
	bsln-emea-euro	1.0M+1.5M	26.2 ± 0.3	32.3 ± 0.1	54.2 ± 0.3
	Random	1.0M+450k	25.5 ± 0.1	31.1 ± 0.1	54.8 ± 0.2
	CE	1.0M+500k	25.4 ± 0.3	31.5 ± 0.3	54.6 ± 0.2
	CRSDS	1.0M+470k	25.8 ± 0.2	31.4 ± 0.2	54.6 ± 0.2
	NNCDS	**1.0**M+**400**k	$\mathbf{25.9 \pm 0.1}$	$\mathbf{31.8 \pm 0.1}$	$\mathbf{54.3 \pm 0.3}$

- DS methods are mostly able to improve the random selection for the language pairs EN-CS and CS-EN. For other language pairs, the random selection has the same behaviour that DS methods but overall, DS method obtains better results.

- Results obtained with the CRSDS and NNCDS methods are slightly better (or similar) than the ones obtained with CE. In some cases are able to reduce the number of sentences significantly.

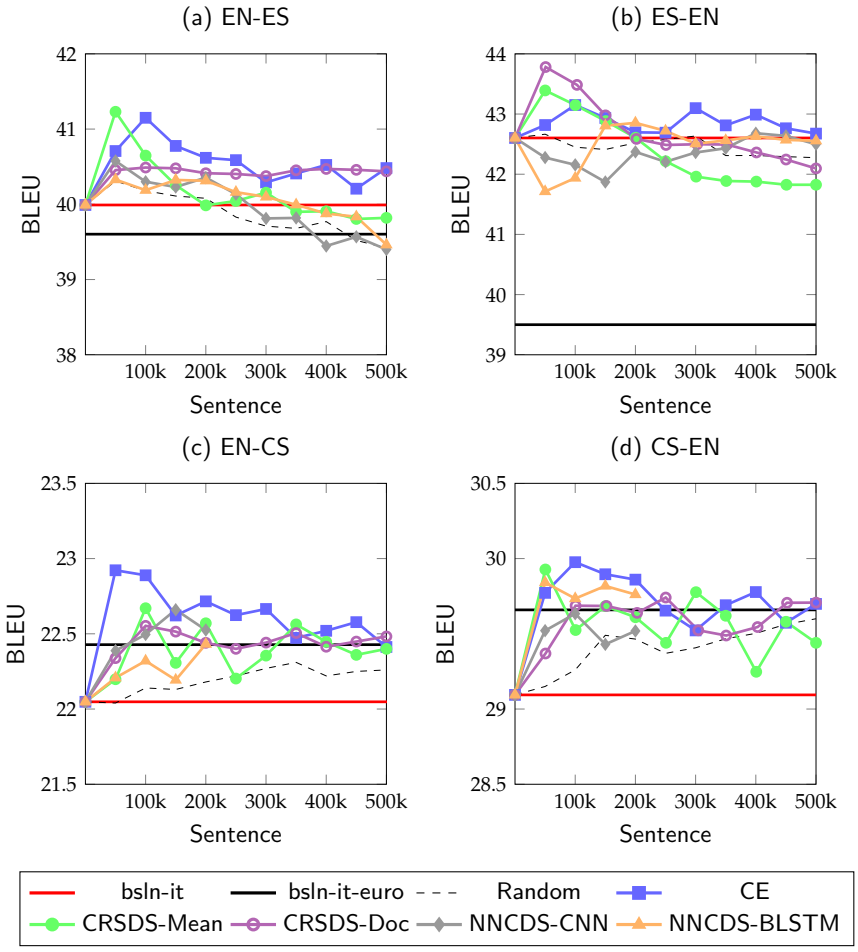


Figure 4.3. Graphical representation of the impact caused on BLEU metric by the addition of sentences to IT domain using monolingual CRSDS, NNCDS, CE, and random selection. Horizontal lines represent bsln-it and bsln-it-euro.

Table 4.6 shows the translation results obtained for IT domain test corpus. In this table, we can see that all DS methods can obtain better translation results than the bsln-it, across language pairs EN-ES, EN-CS and CS-EN. Our DS technique provides better or similar results than the full inclusion of the out-of-domain corpus (bsln-it-euro) across four language pairs, using less than $[10\% - 45\%]$ of the out-

of-domain corpus. This domain is very interesting because the DS method efficacy is proved, in a specific scenario where using all the data available actually decreases the translation quality.

Table 4.6. Best translation results for monolingual DS methods in IT domain. Columns denote, from left to right: Language pairs, DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected and BLEU, METEOR and TER are the evaluation metrics. CE stands for Cross-Entropy method, CRSDS for Continuous Vector-Space Representation of Sentences for Data Selection method and NNCDS for Neural Network Classifier for Data Selection method.

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**
EN-ES	bsln-it	147k	34.9 ± 0.3	59.1 ± 0.2	44.7 ± 0.2
	bsln-it-euro	147k+1.5M	33.1 ± 0.2	58.9 ± 0.2	45.5 ± 0.1
	Random	147k+50k	34.6 ± 0.3	60.0 ± 0.2	44.4 ± 0.2
	CE	147k+100k	35.3 ± 0.3	60.7 ± 0.3	44.1 ± 0.2
	CRSDS	147k+50k	35.4 ± 0.4	60.5 ± 0.3	43.9 ± 0.2
	NNCDS	**147k+50k**	$\mathbf{35.5 \pm 0.2}$	$\mathbf{60.5 \pm 0.3}$	$\mathbf{44.0 \pm 0.3}$
ES-EN	bsln-it	147k	35.3 ± 0.3	37.6 ± 0.1	43.5 ± 0.3
	bsln-it-euro	147k+1.5M	33.1 ± 0.4	37.9 ± 0.2	45.7 ± 0.3
	Random	147k+50k	34.7 ± 0.3	37.7 ± 0.3	44.0 ± 0.3
	CE	147k+100k	34.9 ± 0.2	37.8 ± 0.2	43.5 ± 0.1
	CRSDS	**147k+50k**	$\mathbf{35.1 \pm 0.5}$	$\mathbf{37.9 \pm 0.1}$	$\mathbf{43.3 \pm 0.3}$
	NNCDS	147k+200k	34.6 ± 0.3	38.0 ± 0.1	44.0 ± 0.3
EN-CS	bsln-it	123k	15.9 ± 0.2	23.8 ± 0.1	61.5 ± 0.4
	bsln-it-euro	123k+536k	16.8 ± 0.2	24.7 ± 0.1	59.2 ± 0.3
	Random	123k+350k	16.3 ± 0.6	24.1 ± 0.6	60.7 ± 0.5
	CE	123k+50k	16.8 ± 0.1	24.7 ± 0.1	59.7 ± 0.3
	CRSDS	**123k+100k**	$\mathbf{17.1 \pm 0.1}$	$\mathbf{24.9 \pm 0.1}$	$\mathbf{60.2 \pm 0.6}$
	NNCDS	123k+150k	16.6 ± 0.2	24.7 ± 0.1	60.6 ± 0.5
CS-EN	bsln-it	123k	22.6 ± 0.2	32.0 ± 0.1	55.8 ± 0.2
	bsln-it-euro	123k+536k	23.4 ± 0.2	32.7 ± 0.1	55.0 ± 0.4
	Random	123k+500k	23.6 ± 0.1	32.9 ± 0.1	54.5 ± 0.2
	CE	123k+100k	23.5 ± 0.2	32.4 ± 0.4	55.2 ± 0.8
	CRSDS	**123k+50k**	$\mathbf{23.7 \pm 0.2}$	$\mathbf{32.8 \pm 0.1}$	$\mathbf{54.3 \pm 0.2}$
	NNCDS	123k+50k	23.6 ± 0.3	32.8 ± 0.1	54.3 ± 0.3

4.4.4.2 Bilingual DS method comparative

In this section, we present the results from the comparison of our bilingual DS method with bilingual CE [127]. We discuss the results of

two different domains and across different language pairs. All results were obtained using the development corpus.

Medical domain results

Results comparing our bilingual DS method with bilingual CE (details in Section 3.4.1.1) in the Medical domain are shown in Figure 4.4. In the case of our DS methods, the same approach as in previous section was used. Several conclusions can be drawn:

- Our bilingual DS technique provides better results than including the full out-of-domain corpus (bsln-emea-euro) in language pairs EN-FR, FR-EN, and EN-DE. Specifically, the improvements obtained are in the range of $[0.3 - 0.9]$ BLEU points using less than $[27\% - 19\%]$ of the out-of-domain corpus. In the DE-EN pair our DS strategy does improve the results over the inclusion of the full out-of-domain corpus, but results are very similar using less than 33% of the out-of-domain corpus.
- The results achieved by our bilingual DS strategy are consistently better than those achieved by the bilingual Cross-Entropy method.
- For equal amount of sentences, translation quality is significantly better with the bilingual DS method compared to its monolingual form (Figure 4.2). Hence, the bilingual DS strategy is able to make good use of the bilingual information, reaching a better subset of the out-of-domain data.

Table 4.7 shows the results obtained for the test corpus of Medical domain. As shown, our methods are able to yield competitive results for each language combination. Note that the bilingual methods tend to increase the translation quality and reduce the selected corpus size when compared to monolingual methods (Table 4.5).

4.4.4.3 *Information Technology domain results*

Results comparing our bilingual DS method with bilingual CE (details in Section 3.4.1.1) at IT domain are shown in Figure 4.5. In the

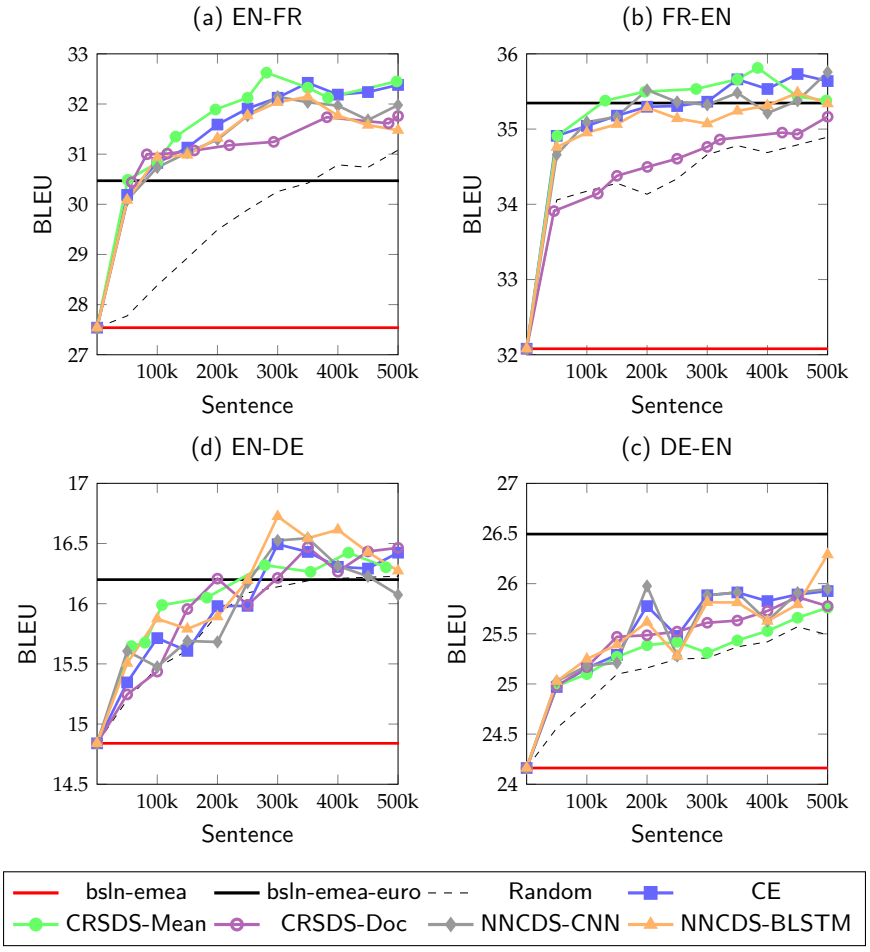


Figure 4.4. Graphical representation of the impact caused on BLEU metric by the addition of sentences to Medical domain using bilingual CRSDS, NNCDS, CE, and random selection. Horizontal lines represent bsln-emea and bsln-emea-euro.

case of our DS methods, the same approach as in the previous section was used. Several conclusions can be drawn:

- Our bilingual DS technique provides better results than baseline bsln-it across the different language pairs. Specifically, the improvements obtained are in the range of $[0.2 - 3.0]$ BLEU points using less than $[4\% - 19\%]$ of the out-of-domain corpus.

Table 4.7. Best translation results for bilingual DS methods in test corpus of Medical domain. Columns denote, from left to right: Language pairs, DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected and BLEU, METEOR and TER are the evaluation metrics. CE stands for Cross-Entropy method, CRSDS for Continuous Vector-Space Representation of Sentences for Data Selection method and NNCDS for Neural Network Classifier for Data Selection method.

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**
EN-FR	bsln-emea	1.0M	28.6 ± 0.1	51.6 ± 0.1	52.7 ± 0.1
	bsln-emea-euro	1.0M+1.5M	29.4 ± 0.1	55.0 ± 0.1	50.2 ± 0.1
	Random	1.0M+500k	29.4 ± 0.3	54.9 ± 0.2	50.4 ± 0.1
	Bili-CE	1.0M+350k	30.2 ± 0.2	55.2 ± 0.2	50.0 ± 0.1
	Bili-CRSDS	**1.0M+281k**	$\mathbf{30.3 \pm 0.2}$	$\mathbf{55.2 \pm 0.1}$	$\mathbf{50.0 \pm 0.2}$
	Bili-NNCDS	1.0M+350k	30.1 ± 0.1	55.1 ± 0.1	50.3 ± 0.4
FR-EN	bsln-emea	1.0M	29.9 ± 0.2	35.4 ± 0.1	48.1 ± 0.2
	bsln-emea-euro	1.0M+1.5M	32.4 ± 0.1	37.6 ± 0.1	45.5 ± 0.1
	Random	1.0M+500k	32.3 ± 0.3	37.4 ± 0.1	45.5 ± 0.2
	Bili-CE	1.0M+450k	32.5 ± 0.2	37.6 ± 0.3	45.4 ± 0.1
	Bili-CRSDS	**1.0M+383k**	$\mathbf{32.8 \pm 0.1}$	$\mathbf{37.8 \pm 0.3}$	$\mathbf{45.3 \pm 0.1}$
	Bili-NNCDS	1.0M+500k	32.5 ± 0.2	37.5 ± 0.2	45.4 ± 0.1
EN-DE	bsln-emea	1.0M	15.6 ± 0.1	38.4 ± 0.2	64.6 ± 0.1
	bsln-emea-euro	1.0M+1.5M	16.6 ± 0.2	40.4 ± 0.2	64.4 ± 0.4
	Random	1.0M+500k	16.6 ± 0.1	40.5 ± 0.2	64.5 ± 0.3
	Bili-CE	1.0M+300k	16.7 ± 0.1	40.2 ± 0.1	63.8 ± 0.4
	Bili-CRSDS	1.0M+350k	16.9 ± 0.2	40.2 ± 0.4	63.7 ± 0.5
	Bili-NNCDS	**1.0M+300k**	$\mathbf{17.1 \pm 0.2}$	$\mathbf{40.9 \pm 0.2}$	$\mathbf{64.2 \pm 0.1}$
DE-EN	bsln-emea	1.0M	23.6 ± 0.2	29.9 ± 0.1	57.1 ± 0.6
	bsln-emea-euro	1.0M+1.5M	26.1 ± 0.1	32.3 ± 0.1	54.2 ± 0.3
	Random	1.0M+450k	25.5 ± 0.1	31.1 ± 0.1	54.8 ± 0.2
	Bili-CE	1.0M+500k	26.0 ± 0.1	31.8 ± 0.2	54.2 ± 0.4
	Bili-CRSDS	1.0M+500k	25.8 ± 0.2	31.6 ± 0.1	54.6 ± 0.4
	Bili-NNCDS	**1.0M+500k**	$\mathbf{26.2 \pm 0.1}$	$\mathbf{32.0 \pm 0.2}$	$\mathbf{54.1 \pm 0.3}$

- Our bilingual DS technique provides better results than including the full out-of-domain corpus bsln-it-euro in language pairs EN-CS and CS-EN. In the case of language pair EN-ES and ES-EN, it is interesting to evidence that increasing the training corpus size does not necessarily produce better translations. In these cases, the efficiency of the DS method is shown. Specifically, the improvements obtained are in the range of $2.7 - 2.9$ BLEU points using less than 4% of the out-of-domain corpus.

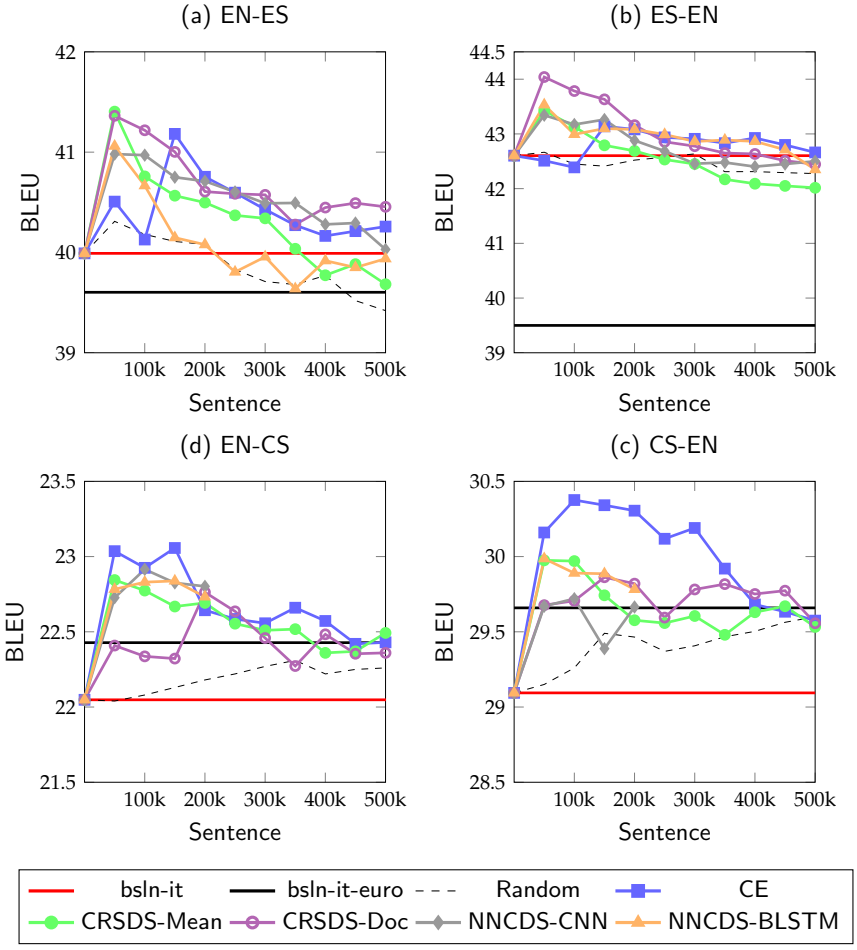


Figure 4.5. Graphical representation of the impact caused on BLEU metric by the addition of sentences to IT domain using bilingual CRSDS, NNCDS, CE, and random selection. Horizontal lines represent bsln-it and bsln-it-euro.

- The results achieved by our bilingual DS strategy are consistently better than those achieved by the bilingual cross-entropy method in the EN-ES and CS-EN language pairs. In the ES-EN and CS-EN cases, results are very similar.

- For equal amount of sentences, translation quality is significantly better with the bilingual DS method, as compared to its mono-

lingual form (Figure 4.5). Hence, the bilingual DS strategy is able to make good use of the bilingual information, reaching a better subset of the out-of-domain data.

Table 4.8 shows the best results obtained for the bilingual Cross-Entropy comparative in this domain in terms of the three evaluation metrics. As shown, our methods are able to yield competitive results for each language combination.

Table 4.8. Best translation results for bilingual DS methods in IT domain. Columns denote, from left to right: Language pairs, DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size,$(+)$ is the number of sentences selected and BLEU, METEOR and TER are the evaluation metrics. CE stands for Cross-Entropy method, CRSDS for Continuous Vector-Space Representation of Sentences for Data Selection method and NNCDS for Neural Network Classifier for Data Selection method.

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**
EN-ES	bsln-it	147k	34.9 ± 0.3	59.1 ± 0.2	44.7 ± 0.2
	bsln-it-euro	147k+1.5M	33.1 ± 0.2	58.9 ± 0.2	45.5 ± 0.1
	Random	147k+50k	34.6 ± 0.3	60.0 ± 0.2	44.4 ± 0.2
	Bili-CE	147k+150k	34.8 ± 0.3	60.4 ± 0.3	44.7 ± 0.1
	Bili-CRSDS	**147k+50k**	$\mathbf{35.9 \pm 0.1}$	$\mathbf{60.5 \pm 0.2}$	$\mathbf{44.5 \pm 0.3}$
	Bili-NNCDS	147k+50k	35.5 ± 0.2	60.2 ± 0.2	44.2 ± 0.2
ES-EN	bsln-it	147k	35.3 ± 0.3	37.6 ± 0.1	43.5 ± 0.3
	bsln-it-euro	147k+1.5M	33.1 ± 0.4	37.9 ± 0.2	45.7 ± 0.3
	Random	147k+50k	34.7 ± 0.3	37.7 ± 0.3	44.0 ± 0.3
	Bili-CE	147k+150k	35.2 ± 0.3	37.9 ± 0.3	43.5 ± 0.2
	Bili-CRSDS	147k+50k	35.5 ± 0.3	38.1 ± 0.1	43.1 ± 0.2
	Bili-NNCDS	**147k+50k**	$\mathbf{35.7 \pm 0.3}$	$\mathbf{38.1 \pm 0.1}$	$\mathbf{43.0 \pm 0.2}$
EN-CS	bsln-it	123k	15.9 ± 0.2	23.8 ± 0.1	61.5 ± 0.4
	bsln-it-euro	123k+536k	16.8 ± 0.2	24.7 ± 0.1	59.2 ± 0.3
	Random	123k+350k	16.3 ± 0.6	24.1 ± 0.6	60.7 ± 0.5
	Bili-CE	123k+150k	17.5 ± 0.2	25.1 ± 0.1	59.1 ± 0.4
	Bili-CRSDS	123k+50k	17.3 ± 0.1	24.8 ± 0.1	59.0 ± 0.2
	Bili-NNCDS	**123k+100k**	$\mathbf{17.5 \pm 0.1}$	$\mathbf{25.1 \pm 0.1}$	$\mathbf{58.7 \pm 0.4}$
CS-EN	bsln-it	123k	22.6 ± 0.2	32.0 ± 0.1	55.8 ± 0.2
	bsln-it-euro	123k+536k	23.4 ± 0.2	32.7 ± 0.1	55.0 ± 0.4
	Random	123k+500k	23.6 ± 0.1	32.9 ± 0.1	54.5 ± 0.2
	Bili-CE	**123k+100k**	$\mathbf{24.0 \pm 0.3}$	$\mathbf{32.5 \pm 0.5}$	$\mathbf{54.9 \pm 1.0}$
	Bili-CRSDS	123k+150k	23.7 ± 0.9	32.8 ± 0.1	54.3 ± 0.3
	Bili-NNCDS	123k+50k	23.9 ± 0.2	32.6 ± 0.1	54.1 ± 0.2

4.4.5 DS METHOD COMPARISON USING THE SOURCE TEST CORPUS

In this section, we now pursue to compare the CRSDS method with the infrequent n-grams method in Section 3.4.2. As exposed in Sections 3.4.2 and 4.2, these methods make the selection process using the source Test corpus. For this reason, in this comparative we do not included NNCDS and CE techniques, because they have not been developed to work in such scenario. The similarity corpus used for CRSDS method (see Section 4.2.1) was the source Test corpus (i.e., $S = T$). All results shown in the figures were obtained with the development corpus with our DS methods.

We compare these two techniques in different domains (Medical and IT).

4.4.5.1 Medical domain results

The results in Figure 4.6 show the effect of adding sentences to the in-domain corpus. In the case of the CRSDS method, the same approach as in the previous section was used. Several conclusions can be drawn:

- Results show that the DS methods yield better results than bsln-emea.
- The results achieved by the CRSDS method are very similar (i.e., not statistically different) from the results achieved by infrequent n-gram recovery in all the languages studied, albeit requiring more sentences.
- Note that for equal amount of sentences added, translation quality with the CRSDS method is very similar when $S = T$ compared to $S = I$ (Figure 4.2), allowing to reduce the number of sentences significantly. We believe that this happens because using the Test corpus entails a better selection of out-of-domain sentences.

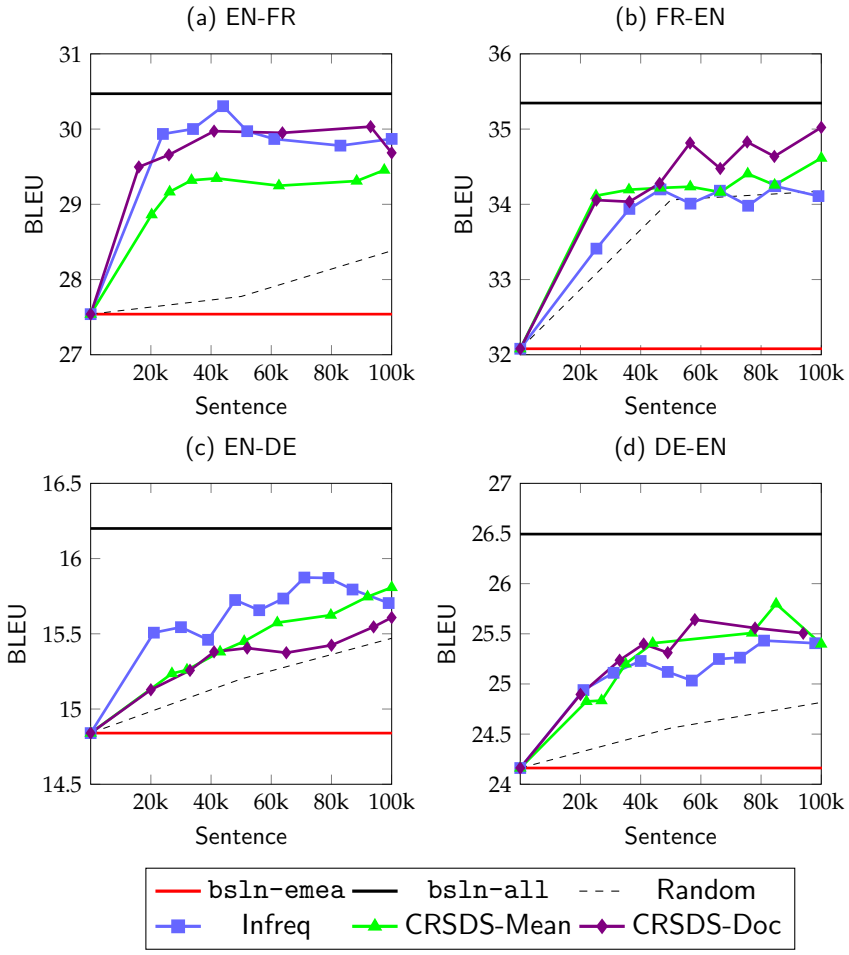


Figure 4.6. Graphical representation of the impact caused on BLEU metric by the addition of sentences to Medical domain using CRSDS, infrequent n-grams recovery, and random DS. Horizontal lines represent the scores of the `bsln-emea` and `bsln-all` systems.

4.4.5.2 *Information Technology domain results*

The results in Figure 4.7 show the effect of adding sentences to the in-domain corpus. Several conclusions can be drawn:

Table 4.9. Best translation results for DS methods using the source test corpus in Medical domain. Columns denote, from left to right: Language pairs, DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected and BLEU, METEOR and TER are the evaluation metrics. CE stands for Cross-Entropy method, CRSDS for Continuous Vector-Space Representation of Sentences for Data Selection method and NNCDS for Neural Network Classifier for Data Selection method.

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**
EN-FR	bsln-emea	1.0M	28.6 ± 0.2	51.6 ± 0.1	52.7 ± 0.1
	bsln-emea-euro	1.0M+1.5M	29.4 ± 0.1	55.0 ± 0.1	50.2 ± 0.1
	Random	1.0M+500k	29.4 ± 0.3	54.9 ± 0.2	50.4 ± 0.1
	Infreq	**1.0M+44k**	$\mathbf{30.2 \pm 0.2}$	$\mathbf{55.2 \pm 0.1}$	$\mathbf{50.0 \pm 0.3}$
	CRSDS	1.0M+41k	29.8 ± 0.2	55.1 ± 0.1	50.0 ± 0.3
FR-EN	bsln-emea	1.0M	29.9 ± 0.2	35.4 ± 0.1	48.1 ± 0.2
	bsln-emea-euro	1.0M+1.5M	32.4 ± 0.1	37.6 ± 0.1	45.5 ± 0.1
	Random	1.0M+500k	32.3 ± 0.3	37.4 ± 0.1	45.5 ± 0.2
	Infreq	**1.0M+85k**	$\mathbf{32.9 \pm 0.1}$	$\mathbf{37.1 \pm 0.1}$	$\mathbf{45.3 \pm 0.1}$
	CRSDS	1.0M+75k	32.6 ± 0.1	37.1 ± 0.1	45.6 ± 0.2
EN-DE	bsln-emea	1.0M	15.4 ± 0.1	38.4 ± 0.2	65.4 ± 0.1
	bsln-emea-euro	1.0M+1.5M	16.6 ± 0.2	40.4 ± 0.2	64.4 ± 0.4
	Random	1.0M+500k	16.6 ± 0.1	40.5 ± 0.2	64.5 ± 0.3
	Infreq	**1.0M+71k**	$\mathbf{16.7 \pm 0.2}$	$\mathbf{39.6 \pm 0.3}$	$\mathbf{63.8 \pm 0.2}$
	CRSDS	1.0M+96k	16.2 ± 0.1	39.4 ± 0.2	64.5 ± 0.3
DE-EN	bsln-emea	1.0M	23.7 ± 0.2	29.9 ± 0.1	57.1 ± 0.6
	bsln-emea-euro	1.0M+1.5M	26.2 ± 0.3	32.3 ± 0.1	54.2 ± 0.3
	Random	1.0M+450k	25.5 ± 0.1	31.1 ± 0.1	54.8 ± 0.2
	Infreq	**1.0M+81k**	$\mathbf{25.8 \pm 0.1}$	$\mathbf{31.6 \pm 0.1}$	$\mathbf{53.9 \pm 0.5}$
	CRSDS	1.0M+94k	25.6 ± 0.1	55.1 ± 0.2	31.6 ± 0.1

- Results show that the DS methods yield better results than bsln-it and bsln-it-euro.
- The results achieved by the CRSDS method are better than the results achieved by infrequent n-gram recovery using the language pairs EN-ES ES-EN and EN-CS. In the case of CS-EN, the results obtained are very similar to the results archived by the infrequent n-gram method.
- Finally, the translation quality with the CRSDS method is significantly better when $S = T$ in comparison to $S = I$ (Figure 4.3), using a smaller number of sentences.

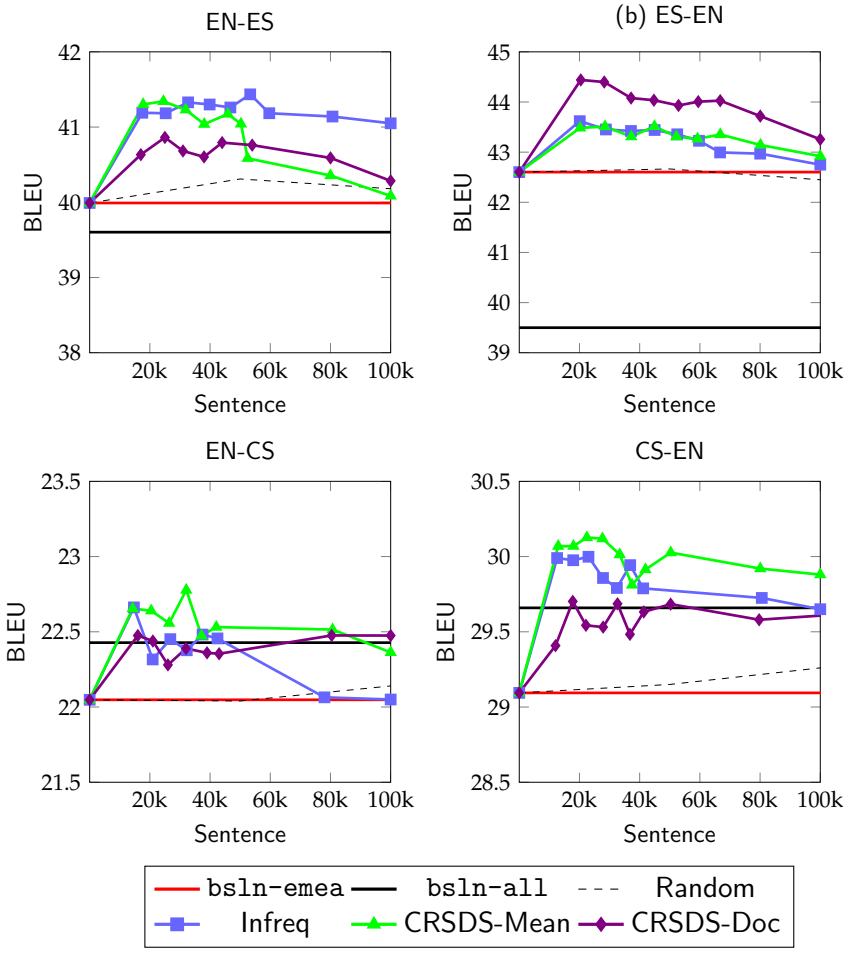


Figure 4.7. Graphical representation of the impact caused on BLEU metric by the addition of sentences to IT domain using CRSDS, infrequent n-grams recovery, and random DS. Horizontal lines represent the score the `bsln-it` and `bsln-it-euro` system.

4.4.5.3 Combination with infrequent n-grams recovery

In this section, we present the experimental results obtained through a re-selection process, in which we use CRSDS method to obtain a first selected corpus, which is then fed as out-of-domain corpus G to the infrequent n-grams method. The ultimate purpose here is to combine

Table 4.10. Best translation results for DS methods using the source test corpus in IT domain. Columns denote, from left to right: Language pairs, DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected and BLEU, METEOR and TER are the evaluation metrics. CRSDS stands for Continuous Vector-Space Representation of Sentences for Data Selection method and Infreq for Infrequent n-grams recovery.

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**
EN-ES	bsln-it	147k	34.9 ± 0.3	59.1 ± 0.2	44.7 ± 0.2
	bsln-it-euro	147k+1.5M	33.1 ± 0.2	58.9 ± 0.2	45.5 ± 0.1
	Random	$147k + 50$k	34.6 ± 0.3	60.0 ± 0.2	44.4 ± 0.2
	Infreq	$147k + 53$k	35.6 ± 0.1	61.0 ± 0.2	44.1 ± 0.2
	CRSDS	**147k+24k**	$\mathbf{35.9 \pm 0.2}$	$\mathbf{60.2 \pm 0.2}$	$\mathbf{44.1 \pm 0.2}$
ES-EN	bsln-it	147k	35.3 ± 0.3	37.6 ± 0.1	43.5 ± 0.3
	bsln-it-euro	147k+1.5M	33.1 ± 0.4	37.9 ± 0.2	45.7 ± 0.3
	Random	147k+50k	34.7 ± 0.3	37.7 ± 0.3	44.0 ± 0.3
	Infreq	147k+20k	35.6 ± 0.4	38.2 ± 0.1	43.0 ± 0.4
	CRSDS	**147k+20k**	$\mathbf{35.9 \pm 0.2}$	$\mathbf{38.0 \pm 0.1}$	$\mathbf{43.0 \pm 0.1}$
EN-CS	bsln-it	123k	15.9 ± 0.2	23.8 ± 0.1	61.5 ± 0.4
	bsln-it-euro	123k+536k	16.8 ± 0.2	24.7 ± 0.1	59.2 ± 0.3
	Random	123k+350k	16.3 ± 0.6	24.1 ± 0.6	60.7 ± 0.5
	Infreq	**123k+15k**	$\mathbf{17.1 \pm 0.2}$	$\mathbf{24.8 \pm 0.1}$	$\mathbf{59.5 \pm 0.5}$
	CRSDS	123k+32k	17.0 ± 0.1	24.9 ± 0.1	59.8 ± 0.5
CS-EN	bsln-it	123k	22.6 ± 0.2	32.0 ± 0.1	55.8 ± 0.2
	bsln-it-euro	123k+536k	23.4 ± 0.2	32.7 ± 0.1	55.0 ± 0.4
	Random	123k+500k	23.6 ± 0.2	32.9 ± 0.1	54.3 ± 0.2
	Infreq	**123k+23k**	$\mathbf{24.2 \pm 0.2}$	$\mathbf{33.2 \pm 0.1}$	$\mathbf{53.7 \pm 0.3}$
	CRSDS	123k+18k	24.0 ± 0.2	32.8 ± 0.1	54.1 ± 0.2

the advantages of both methods, i.e., reducing as much as possible the number of sentences added, while improving translation quality at the same time.

Tables 4.11 and 4.12 show the results obtained for each domain (Medical and IT). In the Medical domain, the combined DS method is able to yield very similar translation quality compared to each DS method individually, but with a much lower amount of sentences. Specifically, the combination is able to reach the same translation quality by adding as few as 1% of the out of domain corpus for EN-FR, 2.5% for DE-EN and 1.6% for EN-DE.

Table 4.11. Summary of the best combination results obtained for each language for Medical domain. Columns denote, from left to right: Language pairs, DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected and BLEU, METEOR and TER are the evaluation metrics. CRSDS stands for Continuous Vector-Space Representation of Sentences for Data Selection method and Infreq for Infrequent n-grams recovery.

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**
EN-FR	bsln-emea	1.0M	28.6 ± 0.2	51.6 ± 0.1	52.7 ± 0.1
	bsln-emea-euro	1.0M+1.5M	29.4 ± 0.1	55.0 ± 0.1	50.2 ± 0.1
	Random	1.0M+500k	29.4 ± 0.3	54.9 ± 0.2	50.4 ± 0.1
	Infreq	1.0M+44k	30.2 ± 0.2	55.2 ± 0.1	50.4 ± 0.3
	CRSDS	1.0M+41k	29.8 ± 0.2	55.1 ± 0.1	50.0 ± 0.3
	CRSDS+Infreq	**1.0M+14k**	$\mathbf{30.0 \pm 0.1}$	$\mathbf{55.3 \pm 0.1}$	$\mathbf{50.0 \pm 0.1}$
FR-EN	bsln-emea	1.0M	29.9 ± 0.2	35.4 ± 0.1	48.1 ± 0.2
	bsln-emea-euro	1.0M+1.5M	32.4 ± 0.1	37.6 ± 0.02	45.5 ± 0.1
	Random	1.0M+500k	32.3 ± 0.3	37.4 ± 0.1	45.5 ± 0.2
	Infreq	1.0M+85k	32.9 ± 0.1	37.4 ± 0.1	45.3 ± 0.1
	CRSDS	1.0M+75k	32.6 ± 0.1	37.1 ± 0.1	45.3 ± 0.1
	CRSDS+Infreq	**1.0M+24k**	$\mathbf{32.9 \pm 0.1}$	$\mathbf{37.1 \pm 0.1}$	$\mathbf{45.4 \pm 0.2}$
EN-DE	bsln-emea	1.0M	15.4 ± 0.1	38.4 ± 0.2	65.4 ± 0.1
	bsln-emea-euro	1.0M+1.5M	16.6 ± 0.2	40.4 ± 0.2	64.4 ± 0.4
	Random	1.0M+500k	16.6 ± 0.1	40.5 ± 0.2	64.5 ± 0.3
	Infreq	1.0M+71k	16.7 ± 0.2	39.6 ± 0.3	63.8 ± 0.2
	CRSDS	1.0M+96k	16.2 ± 0.1	39.4 ± 0.2	64.5 ± 0.3
	CRSDS+Infreq	**1.0M+27k**	$\mathbf{16.7 \pm 0.2}$	$\mathbf{40.6 \pm 0.1}$	$\mathbf{64.2 \pm 0.2}$
DE-EN	bsln-emea	1.0M	23.7 ± 0.2	29.9 ± 0.1	57.1 ± 0.6
	bsln-emea-euro	1.0M+1.5M	26.2 ± 0.3	32.3 ± 0.1	54.2 ± 0.3
	Random	1.0M+450k	25.5 ± 0.1	31.1 ± 0.1	54.8 ± 0.2
	Infreq	1.0M+81k	25.8 ± 0.1	31.6 ± 0.1	53.9 ± 0.5
	CRSDS	1.0M+94k	25.6 ± 0.1	31.6 ± 0.1	55.1 ± 0.3
	CRSDS+Infreq	**1.0M+37k**	$\mathbf{25.9 \pm 0.1}$	$\mathbf{31.8 \pm 0.2}$	$\mathbf{54.1 \pm 0.2}$

In case of the IT domain, the results obtained with the combination technique are better compared to each method individually (language pairs: EN-ES, ES-EN and EN-CS). At the same time, it is important to note that the combination is able to reduce the number of selected sentences to just as few as 3% of the out-of-domain corpus for EN-ES and ES-EN, and 4.1% for EN-CS. In the case of CS-EN the combination obtains very similar translation quality in comparison to each of the methods individually, but is not able to reduce the number of selected sentences.

We consider this specially relevant, since it proves that DS has a very important potential for reducing the computational resources required for training SMT systems.

Table 4.12. Summary of the best combination results obtained for each language for IT domain. Columns denote, from left to right: Language pairs, DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected and BLEU, METEOR and TER are the evaluation metrics. CRSDS stands for Continuous Vector-Space Representation of Sentences for Data Selection method and Infreq for Infrequent ngrams recovery.

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**
EN-ES	bsln-it	147k	34.9 ± 0.3	59.1 ± 0.2	44.7 ± 0.2
	bsln-it-euro	147k+1.5M	33.1 ± 0.1	58.9 ± 0.2	45.5 ± 0.1
	Random	147k+50k	34.9 ± 0.3	60.0 ± 0.2	44.4 ± 0.2
	Infreq	147k+53k	35.6 ± 0.1	61.0 ± 0.1	44.4 ± 0.2
	CRSDS	**147k+24k**	$\mathbf{35.9 \pm 0.2}$	$\mathbf{60.2 \pm 0.2}$	$\mathbf{44.1 \pm 0.2}$
	CRSDS+Infreq	147k+33k	35.5 ± 0.2	59.7 ± 0.1	44.6 ± 0.1
ES-EN	bsln-it	147k	35.3 ± 0.3	37.6 ± 0.1	43.5 ± 0.3
	bsln-it-euro	147k+1.5M	33.1 ± 0.4	37.9 ± 0.2	45.7 ± 0.3
	Random	147k+50k	34.7 ± 0.3	37.7 ± 0.3	44.0 ± 0.3
	Infreq	147k+20k	35.6 ± 0.4	38.2 ± 0.1	43.0 ± 0.4
	CRSDS	147k+20k	35.9 ± 0.2	38.0 ± 0.1	43.0 ± 0.1
	CRSDS+Infreq	**147k+14k**	$\mathbf{36.4 \pm 0.4}$	$\mathbf{38.4 \pm 0.1}$	$\mathbf{42.3 \pm 0.4}$
EN-CS	bsln-it	123k	15.9 ± 0.2	23.8 ± 0.1	61.5 ± 0.4
	bsln-it-euro	123k+536k	16.8 ± 0.2	24.7 ± 0.1	59.2 ± 0.3
	Random	123k+350k	16.3 ± 0.6	24.1 ± 0.6	60.7 ± 0.5
	Infreq	123k+15k	17.1 ± 0.2	24.8 ± 0.1	59.5 ± 0.5
	CRSDS	123k+32k	17.0 ± 0.1	24.9 ± 0.1	59.8 ± 0.5
	CRSDS+Infreq	**123k+22k**	$\mathbf{17.3 \pm 0.2}$	$\mathbf{25.0 \pm 0.1}$	$\mathbf{59.0 \pm 0.2}$
CS-EN	bsln-it	123k	22.6 ± 0.2	32.0 ± 0.1	55.8 ± 0.2
	bsln-it-euro	123k+536k	23.4 ± 0.2	32.7 ± 0.1	55.0 ± 0.4
	Random	123k+500k	23.6 ± 0.2	32.9 ± 0.1	54.3 ± 0.2
	Infreq	123k+23k	24.2 ± 0.2	33.2 ± 0.1	53.7 ± 0.3
	CRSDS	123k+18k	24.0 ± 0.2	32.8 ± 0.1	54.1 ± 0.1
	CRSDS+Infreq	**123k+30k**	$\mathbf{24.4 \pm 0.1}$	$\mathbf{33.1 \pm 0.1}$	$\mathbf{53.4 \pm 0.2}$

4.5 Summary

In this chapter, two novel Data selection techniques have been thoroughly analysed for their application to PBSMT. On the one hand, the theoretical framework for CRSDS and NNCDS techniques have been

proposed. The CRSDS technique is based on the use of a continuous vector-space representation of words or sentences. An important feature of CRSDS is that the selection process is able to make use of an in-domain corpus or a test corpus. The NNCDS technique deals with the problem as a classification task. Following this idea, the NNCDS technique can be seen as a in-domain sentence classifier. Two different neural network architectures were studied: Convolutional neural networks and Bidirectional-Long Short Term Memory networks.

On the other hand, experimental results analysing the effectiveness of such selection procedures have been reported across two different domains and several language pairs. In addition, two different scenarios have been studied where the DS techniques were applied: a scenario in which only in-domain data is available, and a scenario in which the source test corpus is available.

Regarding the selection process when the in-domain corpus is available, results show that CRSDS and NNCDS have an interesting potential in comparison to Cross-Entropy technique. Consistent improvements in translation quality were obtained over different baseline systems with a significant reduction in the number of sentences used to train the SMT system. Results show that our methods are able to obtain better or similar results that the CE method and are able to reduce significantly the number of selected sentences.

Regarding the selection process using the source test corpus, experimental results in the Medical domain with CRSDS reported similar quality compared to the Infrequent n-grams method, albeit it requires more sentences. In the case of the IT domain, the results produced by CRSDS are better when compared to the results achieved by infrequent n-gram recovery. Finally, we proposed a combination of these two techniques, obtaining very good results and a significant reduction of the number of sentences selected. We believe this fact proves the potential behind DS methods.

5 Model combination

* * *

"Muchos años después, frente al pelotón de fusilamiento, el coronel Aureliano Buendía había de recordar aquella tarde remota en que su padre lo llevó a conocer el hielo."

—GABRIEL GARCÍA MÁRQUEZ
CIEN AÑOS DE SOLEDAD

"A lot of years afterwards, in front of the squad of shooting, the colonel Aureliano Buendía had to remember that remote afternoon in that his father carried it to know the ice."

—APERTIUM TRANSLATOR
ONE HUNDRED YEARS OF SOLITUDE
* * *

5.1 Introduction

DS has a positive impact on the translation quality of some domain when we do not have available an in-domain training corpus or when such corpus is not enough. In this chapter, we explore how to make a better use of the data selected by a DS strategy. The selected subset is assumed to contain the sentences from the general corpus that are the most appropriate for improving the translation of the data. In this chapter, we show that the good results previously obtained by DS techniques can be further enhanced by making more intelligent use of the data subset obtained.

More specifically, we explore different combinations of the models trained on the selected subset with the models trained only on the in-

domain corpora. The first method combines the in-domain language model and the selected subset language model by linear interpolation. The second method is based on the combination of translation models (phrase table and reordering table). In the following sections, we explain in detail all these methods. The results show that these combinations lead to improvements over the standard way of using the selected data, namely, concatenating it along with the in-domain data to produce a single SMT model.

The rest of the chapter is organized as follows: Section 5.2 shows a review of state-of-the-art works. Section 5.3 presents the different DS methods and Section 5.4 presents different combination methods. Experiments and discussions are presented in Section 5.5 and the conclusions drawn from the results obtained are presented in Section 5.6.

Table 5.1 shows the abbreviations introduced in the current chapter, in order to facilitate a better comprehension of the text.

Table 5.1. Abbreviations used in Chapter 5.

Abbreviation	Description
SMT	Statistical Machine Translation
DS	Data Selection
CVR	Continuous Vector-space Representation
CE	Cross-Entropy method
Bili-CE	Bilingual cross-entropy method
CRSDS	Continuous Vector-Space Representation of Sentences for DS
Bili-CRSDS	Bilingual CRSDS method
NNCDS	Neural Network Classifier of Sentences for DS
Bili-NNCDS	Bilingual NNCDS method
`Mean`	sentence embedding method
`Doc`	sentence embedding method
CNN	Convolutional Neural Networks
BLSTM	Bidirectional LSTM networks
LM	Language Model
Intr	LM interpolation method
LM+TM	models adaptation by LM interpolation and fill-up method

5.2 Related work

Studies in data selection techniques have typically focused on how to select the best subset of the out-of-domain corpus to concatenate it with the in-domain corpus, and then such concatenation is used to train the final SMT system. In this section, we introduce different approaches available in the literature to combine the in-domain and out-of-domain models. The purpose is to use such approaches for combining the in-domain model with the model trained on the selected data.

In [105] a mixture model approach is proposed. The authors explored different choices: linear and log-linear mixtures. The results show improvements by linear and log-linear mixtures over a baseline trained with all training data.

In [118], the authors proposed to adapt a PBSMT system to new domains by integrating it with language and translation models. Pairs of phrase are here scored with four translation probabilities and four reordering probabilities, thus resulting in a significantly larger set of feature weights to be trained.

In [161] the authors presented their fill-up method, and compare it with standard linear interpolation methods. Given the good results obtained with this method in different research works [162–164], which were in agreement with preliminary results conducted by ourselves, we will use this method. For this reason, the fill-up method will be explained in detail in Section 5.4.2.

In [127] the authors used three methods based in cross-entropy for extracting a pseudo in-domain corpus, detailed in Section 3.4. This pseudo in-domain corpus is used to train a small domain adapted SMT system. The authors combined the small domain-adapted translation model with the true in-domain translation model via linear and log-linear mixtures. In the reported experiments, both mixture methods outperformed the in-domain and general baselines. This work is the most similar to ours, since they explore the interaction between model combination and data selection strategies. However, in this thesis we conduct the study with different DS techniques (see Chap-

ter 4). In addition, we explore language model combination, which was not tackled in the [127] work.

Finally, in [165] a corpus identifier is introduced to distinguish the parallel in-domain corpus from the out-of-domain corpus in a factored translation model. To each target word an id tag is assigned corresponding to the part of the corpus it belongs to. Three additional translation model features are introduced to compute the probability of the corpus id tags being generated given the source phrase, as well as the source and target phrase probabilities, given the corpus id tags. The incorporation of corpus id tags promotes the preference of phrase pairs from a specific domain.

5.3 Data selection method

As introduced in Chapter 3, DS aims to select the best sub-set of bilingual sentences from an available out-of-domain corpus. By doing so, we pretend to improve the translation quality obtained and computational requirements without using the complete pool of sentences.

In this chapter, we decided to apply three different DS techniques for testing this different approach for leveraging the selected-corpus obtained by the DS method. These DS methods were presented in Chapter 3 and Chapter 4, previously.

- Cross-Entropy method (CE), technique details in Section 3.4.1.
- Continuous Vector-Space Representation of Sentences for Data Selection technique (CRSDS), details in Section 4.2.
- Neural Network Classifier for Data Selection technique (NNCDS), details in Section 4.3.

All these techniques are used in their two options: monolingual and bilingual. We referred to monolingual option when used only the source language part of the corpora in the selection process, and the bilingual option is when used all available information (source and target).

5.4 Combination methods

In this section, we present the two different combination methods, Language models linear interpolation (Section 5.4.1) and fill-up method (Section 5.4.2). These two methods have the purpose of capturing in only one model the best part of the two different models trained with different corpora.

5.4.1 LINEAR INTERPOLATION

A common approach to combine multiple language models is to perform a linear interpolation [105], according to the following equation:

$$p(\mathbf{y}) = \sum_c \lambda_c p_c(\mathbf{y}) \tag{5.1}$$

where $p(\mathbf{y})$ refers to the combined language model; $p_c(\mathbf{y})$ is a language model trained on component c and λ_c is the corresponding weight ($\sum_c \lambda_c = 1$).

5.4.2 FILL-UP METHOD

The main idea behind the fill-up method, described in [161], consists in complementing the in-domain phrase table with those phrase pairs of the out-of-domain table that do not appear in the in-domain table.

The fill-up method is applied after a standard PBSMT training process and just before weight optimization. Fill-up effectively exploits background knowledge to improve model coverage, while preserving the more reliable information coming from the in-domain corpus.

Let us assume we have two translation tables (out-of-domain and in-domain corpus): ϕ_G and ϕ_D , with their corresponding phrase translation probabilities $p(\tilde{y}|\tilde{x}, G)$ and $p(\tilde{y}|\tilde{x}, D)$, respectively, where $\tilde{x}$ is a source phrase and $\tilde{y}$ is a target phrase. A fill-up table ϕ_F is defined as follows:

$$\forall(\tilde{x}, \tilde{y}) \in \phi_D \cup \phi_G :$$

$$\phi_F(\tilde{x}, \tilde{y}) = \begin{cases} \{p(\tilde{y}|\tilde{x}, D), \exp(0)\} & \text{if } (\tilde{x}, \tilde{y}) \in \phi_D \\ \{p(\tilde{y}|\tilde{x}, G), \exp(1)\} & \text{otherwise} \end{cases} \tag{5.2}$$

Here, the entries of ϕ_F correspond to the union of the two phrase tables, in which the method considers ϕ_D as the more reliable source and uses it whenever possible. The exponential function (i.e. $\exp(0)$ and $\exp(1)$) is to mark whether a phrase pair is in-domain (ϕ_D) or out-of-domain (ϕ_G). In our experiments, the out-of-domain translation tables ϕ_G is changed as the selected translation tables ϕ_S. This table is calculated using the subset corpus obtained by some DS method.

5.5 Experiments

In this section, we describe the experimental framework employed to assess the performance of the combination methods described above. Then, we show the results obtained with the linear interpolation of language models using the selected set obtained by different DS methods, described in Section 5.5.2. Finally, we present results obtained by combining multiple language models and phrase-tables derived from the use of each DS technique (Section 5.5.3).

5.5.1 Experimental setup

We empirically evaluated different uses of the selected set described in Section 5.4. Here, the intention was to compare the classical use of the selected corpus (see Chapter 4) with the usage proposed in Section 5.4. For this reason, we will employ the same IT domain as in Chapter 4. We conducted the experiments with different language pairs in order to evaluate whether the conclusions drawn from one single language pair holds in further scenarios. The language pairs selected were English-Spanish, Spanish-English, English-Czech and Czech-English. The features of the corpora are shown in Section 2.3. DS methods used the out-of-domain corpus from the Europarl corpus; details of this corpus are shown in Section 2.3. For each DS method, the parameters used are the same as reported in Section 4.4.

The baseline systems are the same as those reported in Chapter 4; bsln-it (obtained by training the SMT system only with in-domain IT training data), bsln-it-euro (obtained by training the SMT system with a concatenation of either the out-of-domain Europarl corpus and the in-domain IT training data). In addition, we also compared the results obtained against another baseline system, bsln-it-LM+TM. In

bsln-it-LM+TM, the language model was created by the interpolation of the in-domain LM and selected LM. The translation models were calculated by the fill-up method ($\phi_D \cup \phi_G$).

In this chapter, SMT output will be evaluated using the automatic metrics: BLEU [80], METEOR [82] and TER [83]. More details about these automatic metrics are given in Section 2.2.

5.5.2 INTERPOLATED LANGUAGE MODEL RESULTS

Using as starting point the positive results obtained with DS techniques, our aim is to make an even better use of the selected subset. In this section, we empirically evaluated the linear interpolation of the language models trained on the in-domain data and the selected subset. We trained two 5-gram language models, one for the in-domain training corpus and another one for the selected subset. Then, these models were interpolated using the SRILM toolkit [97], by computing the combination of weights that best performed on the source side of the test data (using the corresponding source-side language models). Then, such weights were carried over to the target language models.

Figures 5.1 and 5.2 show the main results for each set-up with different DS techniques (monolingual and bilingual options). Results were obtained using the development corpus. In addition, the result obtained with the two baseline systems are also displayed. All results shown in these figures were obtained with the in-domain development corpus. Several conclusions can be drawn:

- Translation quality obtained by interpolating the language model is better in terms of BLEU than the one achieved with the system bsln-it for each language pair.

- Interpolating the language model provides better results than including all the out-of-domain (Europarl) corpus in the SMT system (bsln-it-euro) for all the language pairs. Specifically, the improvements obtained are in the range of $[2.6 - 0.9]$ BLEU points using less than $[70\% - 10\%]$ of the Europarl out-of-domain corpus.

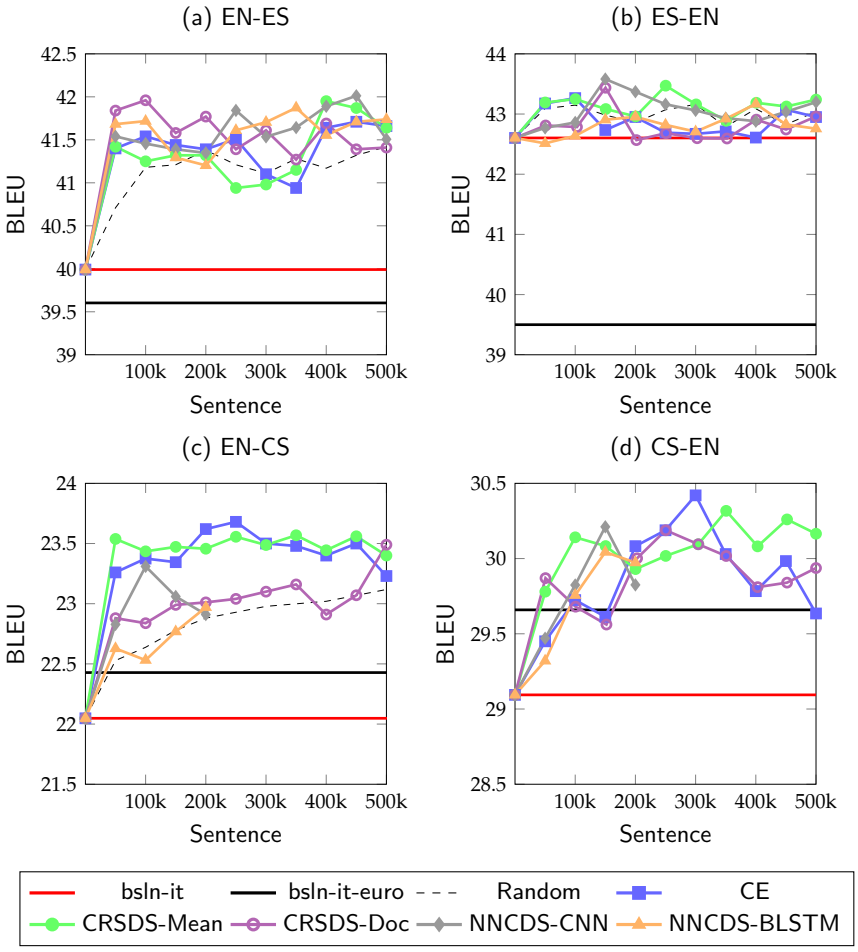


Figure 5.1. Graphical representation of the impact caused on BLEU metric by the use of language model interpolation (in-domain LM and selected subset from the Europarl out-of-domain corpus) for each monolingual DS method.

- Results do not show significant differences between monolingual or bilingual DS methods.

Table 5.2 presents the test corpus results obtained using different DS methods to select an appropriate subset for language model interpolation. Results are presented in terms of BLEU. For each DS method, results of the monolingual and bilingual options are given.

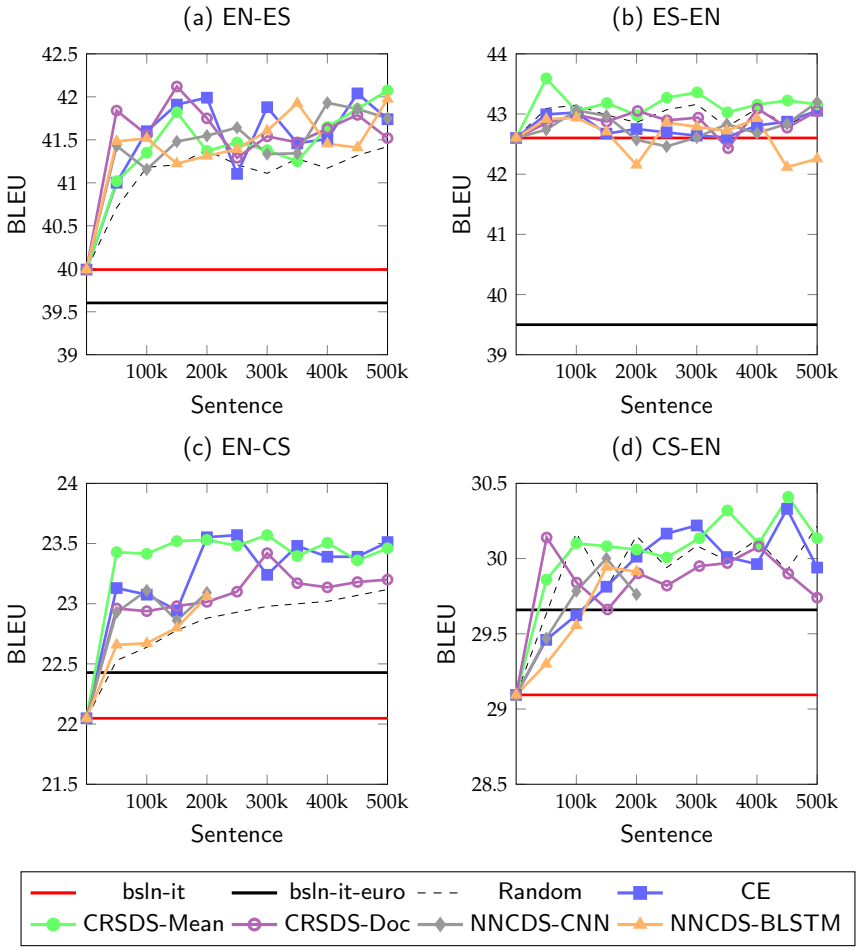


Figure 5.2. Graphical representation of the impact caused on BLEU metric by the use of language model interpolation (in-domain LM and selected subset from the Europarl out-of-domain corpus) for each bilingual DS method.

In addition, we show the baseline system and random selection results. Some conclusions can be drawn:

- As shown, making the linear interpolation using the selected subset of the out-of-domain corpus using any of the DS techniques achieves a better performance when compared to bsln-it and bsln-it-euro. We believe that this is because the DS tech-

niques are able to select more relevant sentences from the out-of-domain corpora.

- The selected corpus used to make linear interpolation of the LM yielded better translation results than the random method, reducing also the corpus size.

- In general, the results show no difference between the DS methods in terms of the automatic metrics. The major difference is the size of the selected subset.

5.5.3 TRANSLATION MODEL COMBINATION RESULTS

In addition to the language model interpolation, we also wanted to adapt the phrase table. For both SMT systems, we trained a standard PBSMT system and applied the fill-up method described in Section 5.4.2, combining both phrase and reordering tables. This combination of models will be identified by LM+TM

Figures 5.3 and 5.4 show the main results for each set-up with different DS techniques (monolingual and bilingual options). In addition, the results obtained with the three baseline systems are also displayed. Several conclusions can be drawn:

- Translation quality obtained by combining the phrase-table is better in terms of BLEU than the one achieved with bsln-it and bsln-it-euro for each language pair.

- Results show than it is better to combine the translation model using the best sub-set rather than using all the available data.

Table 5.3 presents the test corpus results obtained using different DS methods to select an appropriate subset in order to combine the translation models by the LM+TM method. Results are presented in terms of the three different automatic metrics. For each DS method, results are given for monolingual and bilingual options. In addition, we show the results of the baseline system and the random selection. Some conclusions can be drawn:

Table 5.2. Best translation results for the IT domain test corpus when using language model interpolation (Intr) and different DS techniques. Columns denote, from left to right: DS methods, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected, and BLEU the evaluation metric. CE stands for Cross-Entropy method, CRSDS for Continuous Vector-Space Representation of Sentences for Data Selection method and NNCDS for Neural Network Classifier for Data Selection method.

	EN-ES		**ES-EN**	
Method	$\|S\|$	**BLEU**	$\|S\|$	**BLEU**
bsln-it	147k	34.9 ± 0.3	147k	35.3 ± 0.3
bsln-it-euro	147k+1.5M	33.1 ± 0.2	147k+1.5M	33.1 ± 0.4
Random-Intr	147k+500k	35.4 ± 0.1	147k+300k	35.7 ± 0.2
CE-Intr	147k+450k	35.7 ± 0.1	147k+100k	35.6 ± 0.1
CRSDS-Intr	147k+400k	35.9 ± 0.1	147k+250k	35.7 ± 0.2
NNCDS-Intr	147k+450k	36.0 ± 0.3	$147k + 150k$	35.8 ± 0.2
Bili-CE-Intr	147k+300k	35.9 ± 0.1	147k+500k	35.6 ± 0.2
Bili-CRSDS-Intr	**147k+150k**	$\mathbf{36.0 \pm 0.1}$	**147k+50k**	$\mathbf{36.0 \pm 0.1}$
Bili-NNCDS-Intr	147k+500k	35.9 ± 0.3	$147k + 100k$	35.6 ± 0.2
	EN-CS		**CS-EN**	
bsln-it	123k	15.9 ± 0.2	123k	22.6 ± 0.2
bsln-it-euro	123k+536k	16.8 ± 0.2	123k+536k	23.4 ± 0.2
Random-Intr	123k+400k	16.4 ± 0.3	123k+500k	23.2 ± 0.2
CE-Intr	**123k+250k**	$\mathbf{16.6 \pm 0.1}$	123k+300k	23.3 ± 0.2
CRSDS-Intr	123k+350k	16.6 ± 0.1	123k+350k	23.3 ± 0.2
NNCDS-Intr	123k+100k	16.4 ± 0.1	**123k+150k**	$\mathbf{23.3 \pm 0.1}$
Bili-CE-Intr	123k+250k	16.6 ± 0.1	123k+450k	23.2 ± 0.1
Bili-CRSDS-Intr	123k+300k	16.6 ± 0.1	123k+450k	23.2 ± 0.2
Bili-NNCDS-Intr	123k+200k	16.4 ± 0.1	123k+200k	23.1 ± 0.2

- The application of the fill-up method using the selected subset of the out-of-domain corpus using any of the DS techniques achieves a better performance when compared to bsln-it, bsln-it-euro and bsln-LM+PB. We believe that this takes place because the DS techniques are able to select more relevant sentences from the out-of-domain corpora.

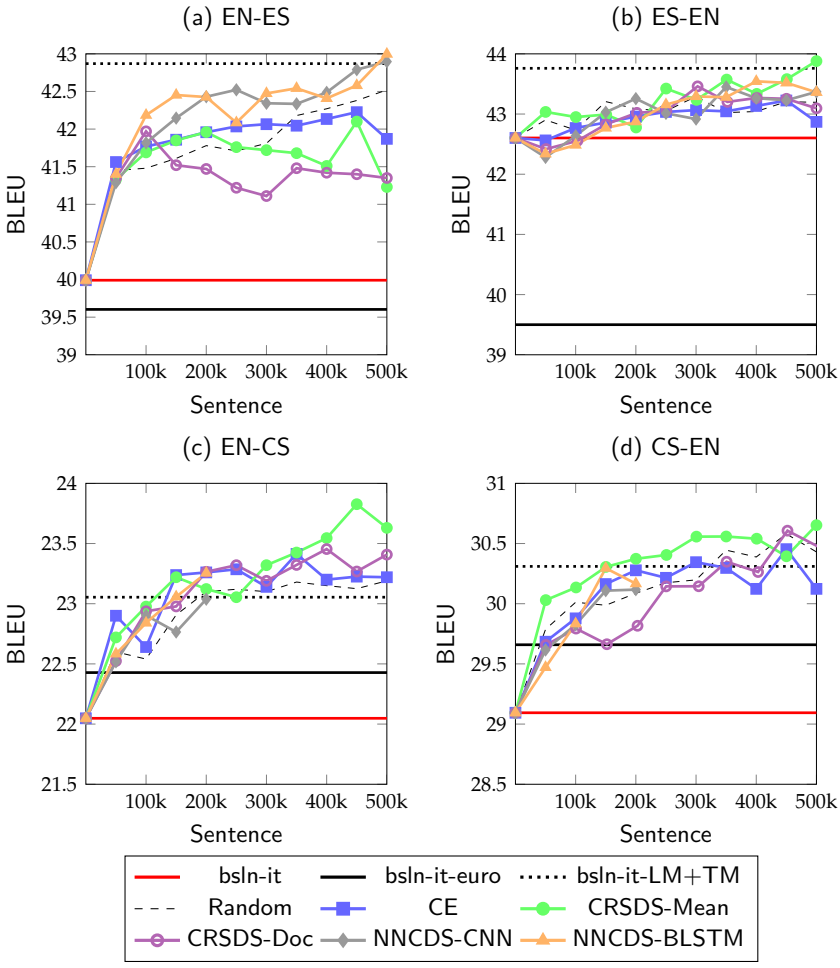


Figure 5.3. Graphical representation of the impact caused on BLEU metric by the use of language model interpolation (in-domain LM and selected subset from the out-of-domain corpus) and phrase-based combination for each monolingual DS techniques. Horizontal lines represent the baselines scores using the in-domain corpus, all data available, and the combined-model baseline (bsln-it-LM+TM).

- The selected corpus yielded better translation results than the random method, reducing also the corpus size.

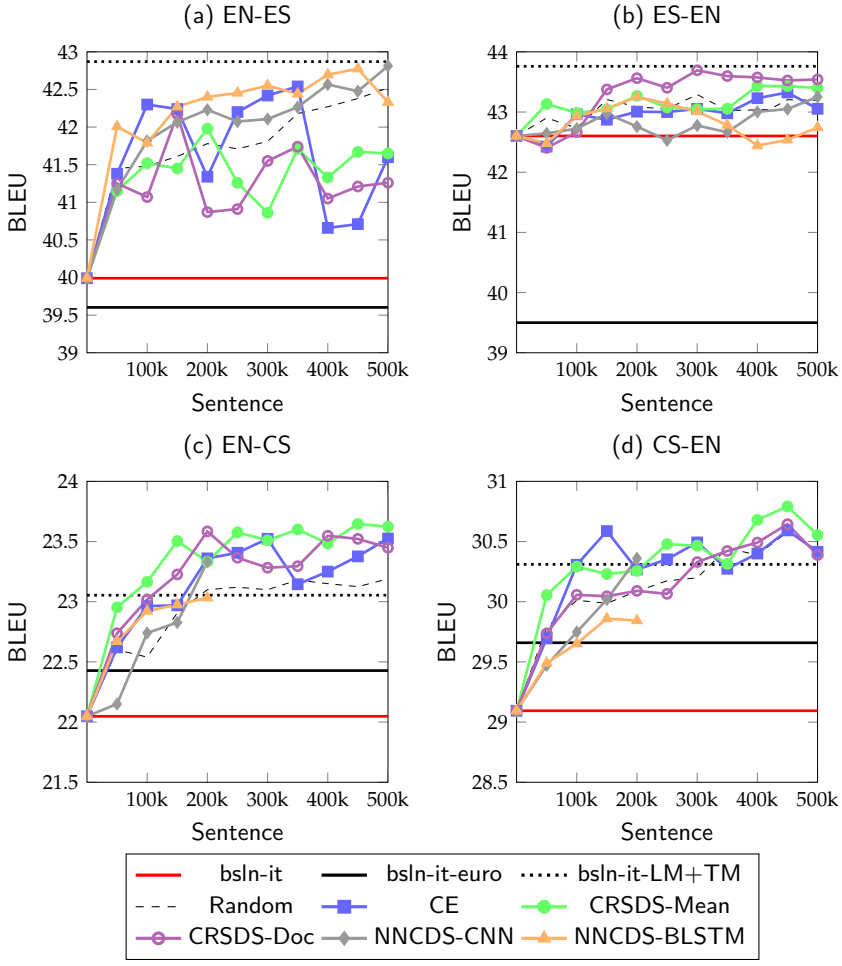


Figure 5.4. Graphical representation of the impact caused on BLEU metric by the use of language model interpolation (in-domain LM and selected subset from the out-of-domain corpus) and phrase-based combination for each bilingual DS techniques. Horizontal lines represent the scores of the baselines systems.

- The results show no difference between the DS methods in terms of automatic metric. The major difference is the size of the selected subset.

Table 5.3. Best translation results for the IT domain applying LM interpolation and translation models combination. Columns denote, from left to right: DS methods, combination method, $|S|$ stands for the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected, and the evaluation metric BLEU. CE stands for Cross-Entropy method, CRSDS for Continuous Vector-Space Representation of Sentences for Data Selection method and NNCDS for Neural Network Classifier for Data Selection method.

	EN-ES		**ES-EN**	
System	$\|S\|$	**BLEU**	$\|S\|$	**BLEU**
bsln-it	147k	34.1 ± 0.1	147k	34.1 ± 0.1
bsln-LM+PB	147k+1.5M	36.6 ± 0.3	147k+1.5M	36.5 ± 0.1
Random-LM+PB	147k+500k	35.4 ± 0.1	147k+500k	35.4 ± 0.1
CE-LM+PB	147k+450k	36.6 ± 0.1	147k+450k	36.2 ± 0.2
CRSDS-LM+TM	147k+450k	36.0 ± 0.3	**147k+500k**	$\mathbf{36.9 \pm 0.3}$
NNCDS-LM+TM	**147k+500k**	$\mathbf{37.0 \pm 0.1}$	147k+400k	36.5 ± 0.2
Bili-CE-LM+PB	147k+350k	36.6 ± 0.2	147k+450k	36.3 ± 0.4
Bili-CRSDS-LM+TM	147k+150k	36.2 ± 0.3	147k+300k	36.7 ± 0.3
Bili-NNCDS-LM+TM	147k+500k	36.8 ± 0.1	147k+200k	36.3 ± 0.3
	EN-CS		**CS-EN**	
bsln-it	123k	15.2 ± 0.2	123k	22.6 ± 0.2
bsln-LM+TM	123k+536k	17.3 ± 0.2	123k+536k	24.5 ± 0.1
Random-LM+TM	123k+500k	17.2 ± 0.2	123k+450k	24.5 ± 0.2
CE-LM+TM	123k+350k	17.4 ± 0.1	123k+450k	24.5 ± 0.2
CRSDS-LM+TM	**123k+450k**	$\mathbf{17.8 \pm 0.2}$	123k+500k	24.7 ± 0.1
NNCDS-LM+TM	123k+200k	17.3 ± 0.1	123k+150k	24.3 ± 0.2
Bili-CE-LM+PB	123k+300k	17.5 ± 0.1	123k+450k	24.6 ± 0.1
Bili-CRSDS-LM+TM	123k+200k	17.6 ± 0.2	**123k+450k**	$\mathbf{24.8 \pm 0.2}$
Bili-NNCDS-LM+TM	123k+200k	17.3 ± 0.2	123k+200k	24.4 ± 0.2

5.5.4 Comparison with a concatenation approach

In this section, a comparison of the different use of the selected subset is presented. For this purpose, the best results from Chapter 3 are shown again. Table 5.4 presents the translation quality results in terms of all three metrics and obtained with the different methods used for each combination of the in-domain and out-of-domain corpora (Intr and LM+TM) and the baseline system. We compare

the results obtained in this chapter with the best results archived for each language in the previous chapter (see Sections 4.4.4.1 and 4.4.4.3). Also, training computational time is reported for each system in this section. All the experiments were performed under the same conditions, using 64 bit machine with Intel Xeon CPUs at 2.50 GHz with 6 MB cache. Several conclusions can be drawn:

- As shown, training the SMT system on a selected subset of the out-of-domain corpora (Europarl corpus) using some DS technique achieves a performance improvement of around $[+0.2 - 1.8]$ BLEU points, $[+0.8 - 1.5]$ METEOR and $[-0.1 - 2.0]$ TER points, when compared to the bsln-it and bsln-it-euro baselines. We believe that this is because the DS techniques are able to select more relevant sentences from the out-of-domain corpora.

- All the combination techniques used (language model interpolation and phrase table combination) yield improvements over the two baseline systems. This demonstrates that combining the models is a better option than using all available data for training.

- Results obtained in terms of the different metrics with language model interpolation are able to improve (or achieve similar results) further over the system trained on both in-domain data and the selected subsets (DS) with a significant reduction of the computational time for the language pair EN-ES and ES-EN. We think these are good examples that show the effectiveness of applying an interpolation method over the selected subset.

- In addition, the results for the test corpus are shown for each language pair were obtained with the combination of linear interpolation and model combination (DS-LM+TM). These results are able to improve further over the bsln-LM+TM system, achieving an additional BLEU and METEOR increase of around 0.4 points, and additional TER decreases of around $[0.2 - 0.5]$ points, with a significant reduction in the computational resources required. We understand this is because these two methods are able to make a better use of the selected subset.

- Computational time evidences that a correct selection of the data can also lead to a reduction of the time required to train the system.

Table 5.4. Summary of the best results obtained with each set-up. Columns denote, from left to right: Language, SMT systems, $|S|$ is the number of sentences which is given in terms of the in-domain corpus size, $(+)$ is the number of sentences selected, BLEU, METEOR, TER evaluation metrics and computational time spent (minutes).

Language	System	$\|S\|$	**BLEU**	**METEOR**	**TER**	**Time**
EN-ES	bsln-it	147k	34.1 ± 0.1	58.4 ± 0.1	45.5 ± 0.1	190
	bsln-it-euro	147k+1.5M	33.4 ± 0.1	59.2 ± 0.1	45.3 ± 0.2	1580
	bsln-LM+TM	147k+1.5M	36.6 ± 0.3	61.4 ± 0.2	42.7 ± 0.3	1105
	DS	147k+50k	36.1 ± 0.5	60.7 ± 0.5	43.7 ± 0.5	300
	DS-Intr	147k+150k	36.0 ± 0.1	60.3 ± 0.1	43.6 ± 0.3	255
	DS-LM+TM	**147k+500k**	$\mathbf{37.0 \pm 0.1}$	$\mathbf{61.4 \pm 0.1}$	$\mathbf{42.5 \pm 0.1}$	878
ES-EN	bsln-it	147k	35.3 ± 0.3	37.6 ± 0.1	43.5 ± 0.3	200
	bsln-it-euro	147k+1.5M	33.6 ± 0.4	37.9 ± 0.2	44.8 ± 0.5	1450
	bsln-LM+TM	147k+1.5M	36.5 ± 0.1	38.4 ± 0.1	42.2 ± 0.1	1240
	DS	147k+50k	35.5 ± 0.2	38.1 ± 0.1	43.1 ± 0.2	285
	DS-Intr	147k+150k	36.0 ± 0.1	37.7 ± 0.1	43.0 ± 0.1	270
	DS-LM+TM	**147k+500k**	$\mathbf{36.9 \pm 0.3}$	$\mathbf{38.7 \pm 0.1}$	$\mathbf{41.8 \pm 0.1}$	855
EN-CS	bsln-it	123k	15.2 ± 0.2	23.5 ± 0.1	61.7 ± 0.3	180
	bsln-it-euro	123k+536k	15.7 ± 0.4	24.0 ± 0.5	61.8 ± 0.4	620
	bsln-LM+TM	123k+536k	17.3 ± 0.2	24.3 ± 0.2	59.5 ± 0.2	520
	DS	123k+100k	17.5 ± 0.1	25.1 ± 0.1	58.6 ± 0.4	385
	DS-Intr	123k+250k	16.6 ± 0.1	24.2 ± 0.1	60.5 ± 0.3	220
	DS-LM+TM	**123k+450k**	$\mathbf{17.8 \pm 0.2}$	$\mathbf{24.8 \pm 0.1}$	$\mathbf{58.7 \pm 0.1}$	410
CS-EN	bsln-it	123k	22.6 ± 0.2	32.0 ± 0.1	55.8 ± 0.2	170
	bsln-it-euro	123k+536k	23.4 ± 0.2	32.7 ± 0.1	55.0 ± 0.4	600
	bsln-LM+PB	123k+536k	24.5 ± 0.1	32.9 ± 0.1	53.5 ± 0.2	520
	DS	123k+100k	24.0 ± 0.3	32.3 ± 0.5	54.9 ± 1.0	398
	DS-Intr	123k+150k	23.3 ± 0.1	32.3 ± 0.1	55.2 ± 0.1	203
	DS-LM+TM	**123k+450k**	$\mathbf{24.8 \pm 0.2}$	$\mathbf{33.5 \pm 0.1}$	$\mathbf{53.0 \pm 0.1}$	425

5.6 Summary

Data selection has received an increasing amount of interest within the SMT research community. In this chapter, we studied different uses of the bilingual sentences selected with different DS methods. We proposed to combine the language and translation models estimated on the in-domain data with those estimated on the selected

subsets. First, we proposed to interpolate the language model (in-domain LM and subset LM). Second, we proposed to combine the phrase tables (both translation tables and reordering tables). In this case we used a fill-up method to obtain the new tables. The results coming from different combinations show improvements in terms of different automatic metrics with respect to a system trained on all the data available. In addition, a reduction of the computational time required to train the system is achieved.

6 Looking for the right development corpus

* * *

"My love for Heathcliff resembles the eternal rocks beneath: a source of little visible delight, but necessary."

—EMILY BRONTË
WUTHERING HEIGHTS

"Mi amor por Heathcliff se asemeja a las rocas eternas que sobresalen profundamente enterradas en la tierra: son motivo de escaso goce para quien las contempla, pero al mismo tiempo son necesarias."

—HUMAN TRANSLATOR
CUMBRES BORRASCOSAS

* * *

6.1 Introduction

As explained in Chapter 1, the tuning process is a critical step in every system that presents a weighted combination of features. It adjusts the weights so that they best fit the target distribution. This process typically yields important improvements in the performance of the system developed. However, selecting an appropriate development set is crucial for this process to reach its goal. In [166], their experiments show that using different development corpora to optimize the log-linear weights of a SMT system, the results can vary up to 2.5 BLEU points. For this reason, obtaining a good development corpus is an important task in SMT.

The DS task is stated as the problem of selecting the best subcorpus of sentences from an available pool of sentences used to train

a machine learning system. This chapter deals with DS, but here the aim is to select, out of an available pool of sentences the best development corpus for a given test set using a log-linear weight optimization. With this purpose, these methods focus on creating an appropriate development corpus to achieve better translation quality on a given test set, particularly, when hand-crafted development sets are not available.

We study our development DS techniques in two different tasks. In the first case, the purpose is to analyse the behaviour of our techniques in a controlled scenario where the data is labelled according to domain. The goal is to study our method capacity of properly predict the domain labels together with the translation quality achieved. In the second scenario, we evaluate the techniques presented in a real task, where a specific test set belonging to the texts of a real e-commerce site is provided (without domain labels).

This chapter is organised as follows: Section 6.2 briefly lists other works dealing with related issues, both focused on finding and selecting the most suitable development corpus. The different methods proposed in the present work for creating the best development corpus are described in Section 6.3. Then, the experimental results obtained for each task are described in Section 6.4.2 and Section 6.4.3. Lastly, Section 6.5, describes the conclusions drawn from the present work.

Table 6.1 shows the abbreviations introduced in the current chapter, in order to facilitate a better comprehension of the text.

Table 6.1. Abbreviations used in Chapter 6.

Abbreviation	Description
SMT	Statistical Machine Translation
DS	Data Selection
DDS	Development Data Selection
LD	Levenshtein distance
CVR	Continuous Vector-space Representation
TF	TF-IDF method
`Doc`	sentence embedding method

6.2 Related work

The work presented here is close in concept to the domain adaptation scenario. Different domain adaptation techniques, including data selection and mixture models have been developed for different scenarios. A wide variety of data selection methods have been used over the years. The main principle is to measure the similarity of sentences from the out-of-domain corpus to some in-domain corpus, either the development or the (source side of the) test set. Such similarity is often based on information theory metrics as, perplexity or cross entropy. In this thesis, we dedicate other Chapters to DS methods, details can be found in Chapters 2 and 3.

DS approaches assume that the selection corpus is used to train or combine the SMT models. However, there are evidence of research about the selection of the appropriate development corpus. Such research can be split into two categories: transductive and inductive learning. In the first category, a development set is chosen, from among several "closed" development sets, based on the test set at hand [167–169]. The second category deals with the problem without knowing the test set beforehand, but knowing the domain of the test set. Previous work on development data selection for unknown test sets include [138, 166, 170]. Note that, our work has an important difference regarding both, transductive and inductive learning. Even though it is closer to the transductive learning setting; all these works are based on the selection of the most adequate development corpus from a collection of "closed" development corpora, choosing the one that belongs to the test set domain. In our case, we want to construct a specific development corpus for a given test corpus without knowing the domain of the test set.

6.3 Development DS techniques

The main idea behind the Development Data Selection (DDS) is to create the best development corpus from an available pool of sentences, given a specific source test set, when the in-domain development corpus is not available. In this section, we present three different DDS methods following different criteria.

6.3.1 LEVENSHTEIN DISTANCE DDS

The first DDS technique proposed involves computing the edit distance (Levenshtein Distance) between a candidate sentence and the closest sentence in the test set. Here, the intuition is to consider that a given sentence is a good candidate to be included in the development set if it is not too far away from the sentences in the test set T, as measured by the Levenshtein Distance. We will refer to this technique as LD-DDS .

The Levenshtein Distance (LD) [171] is a string metric for measuring the difference between two sequences (words or sentences). The LD between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to make them match.

Algorithm 4 shows the procedure. P is the pool of sentences available, $[\mathbf{x}_p, \mathbf{y}_p]$ is an out-of-domain sentence pair ($[\mathbf{x}_p, \mathbf{y}_p] \in P$), and $|P|$ is the number of sentences in P. Our objective is to select the most suitable data from P such that it is (for translating) belonging to the test corpus T (composed only by source sentences). In this way, an appropriate development corpus (`Dev-Corpus`) is created.

Data: pool P; test data T; threshold τ
Result: Development corpus `Dev-Corpus`

```
forall t in T do
    forall [x_p, y_p] in P do
        if LD(t, x_p) ≤ τ then
            if [x_p, y_p] ∉ Dev-Corpus then
                add [x_p, y_p] to Dev-Corpus
                remove [x_p, y_p] to P
            end
        end
    end
end
```

Algorithm 4: Pseudo-code for LD-DDS.

Algorithm 4 introduces the $LD(\cdot, \cdot)$ function, which computes the LD between two given sentences. Note that, threshold τ establishes

the size of the development corpus, and will need to be fixed empirically (Section 6.4.2).

6.3.2 DDS WITH VECTOR-SPACE REPRESENTATIONS

We present two other DDS selection techniques, where the common point is that they both leverage a continuous vector-space representation of the sentences involved. First, we will describe our technique in abstract terms, and then we will present two different candidates for obtaining a continuous vector-space representation $F(\mathbf{x})$ (or $F_{\mathbf{x}}$ for short) of a given sentence $\mathbf{x}$.

The intuition is to select as candidate sentences those whose vector-space representation is similar to those in the test set, assuming that similar sentences will have similar vectors.

The advantage of having a continuous vector-space representation of the test sentences is that a mean can be computed. It can be assumed as a sort of sentences prototype present in the test set. It was not possible to compute the mean in the case of LD-DDS (Section 6.3.1).

Probably, the best way to explain this intuition is graphically (Figure 6.1). This Figure is a graphical example of the idea that we follow in this section. Sentences are represented in a two-dimensional vector-space. Blue points are the representation of the test sentences and red points represent the vectors of the sentences of the available pool of sentences, from which the development set is to be selected. Assuming that similar sentences will have a similar vector-space representation, the vectors of the test corpus will be very close to each other whereas the vectors for the general pool of sentences will be more disperse. The idea of our method is to draw a circle boundary, containing all test-sentences and (hopefully) only a few of the sentences in the candidate pool. The radius of this circumference (or hyper-sphere in a multi-dimensional vector-space) is established as the distance between the center of the test set (mean) and the furthest sentence of the test set.

Algorithm 5 shows the procedure. Here, P is the pool of candidate sentences, $[\mathbf{x}_p, \mathbf{y}_p]$ is a candidate sentence pair, with $[\mathbf{x}_p, \mathbf{y}_p] \in P$,

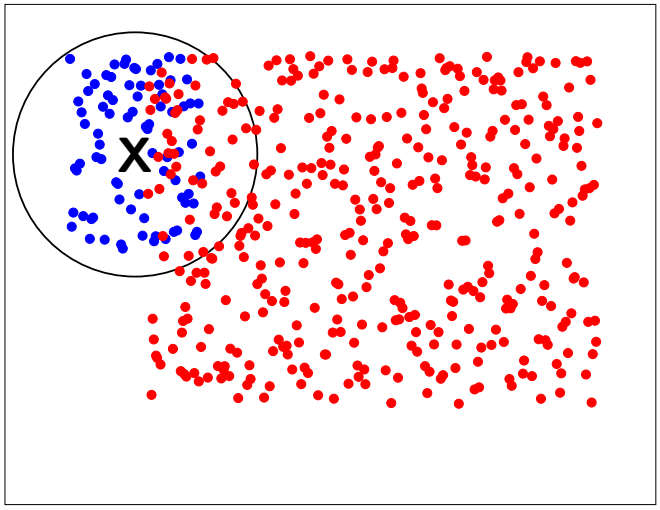


Figure 6.1. Graphical representation of the intuition behind our vector-space selection techniques. Red points represent the development sentence vectors, blue points represent the test sentence vectors. X is the mean of the test vectors and the circumference represents the boundary obtained.

$F_{\mathbf{x}}$ is the vector-space representation of $\mathbf{x}$, and $|P|$ is the number of sentences in P. Then, our objective is to select the most suitable data from P belonging to the source test data T. For this purpose, we define $F_{\mathbf{t}}$ as the vector-space representation of a sentence $\mathbf{t} \in T$.

Algorithm 5 introduces several functions:

- $mean(\cdot)$: calculates mean $F_{\mathbf{T}}$ for test corpus T, assuming a *size* dimensional vector-space:

$$F_{\mathbf{T}} = \frac{1}{|T|} \sum_{t}^{|T|} F_{\mathbf{t}} \tag{6.1}$$

- $cos(\cdot, \cdot)$: the cosine similarity between two different vectors, e.g.:

$$cos(F_{\mathbf{t}}, F_{\mathbf{T}}) = \frac{F_{\mathbf{t}} \cdot F_{\mathbf{T}}}{\|F_{\mathbf{t}}\| \cdot \|F_{\mathbf{T}}\|} \tag{6.2}$$

In addition, τ represents the radius of the circumference, which is computed in lines 2 to 6 (the first **forall** loop) in Algorithm 5.

Data: Pool P; test data T
Result: Development corpus `Dev-Corpus`

```
F_T = mean(T);
τ = 0;
forall t in T do
    if cos(F_t, F_T) ≥ τ then
        τ = cos(F_t, F_T)
    end
end
forall [x_p, y_p] in P do
    if cos(F_{x_p}, F_T) ≤ τ then
        add [x_p, y_p] to Dev-Corpus
        remove [x_p, y_p] to P
    end
end
```

Algorithm 5: Pseudo-code for DDS leveraging vector-space representations of sentences.

Once the selection algorithm has been established, now we need to define how to represent sentences in a Z-dimensional space. Using vector-space representation for textual data (word, sentence or document) is not a new idea and has been widely used in a variety of NLP applications. These representations have recently demonstrated promising results across a variety of tasks [42, 45, 53, 172–174], such as speech recognition, part-of-speech tagging, sentiment classification and identification, information retrieval and machine translation.

We have used two different approaches for representing sentences in a continuous vector-space: the popular *term frequency – inverse document frequency* (TF-IDF) [175], and sentence embeddings [49] (previously used in Chapter 3 and Chapter 4). The basic idea is to represent a sentence $\mathbf{x}$ with a real-valued vector of some fixed dimension *size*, that is able to capture similarity (lexical, semantic or syntactic) between a given pair of sentences.

6.3.2.1 TF-IDF representation

The TF-IDF values can be used to create vector representations of sentence or documents. Using this kind of representation in a common vector-space is called vector space model [176], which is not only used in information retrieval but also in a variety of other research fields like machine learning (e.i. clustering, classification).

Each sentence $\mathbf{x} \in P$ is represented as a numeric vector $F_{\mathbf{x}} = (F_{x_1}, \ldots, F_{x_k}, \ldots, F_{x_{|V|}})$, where $|V|$ is the size of the vocabulary V. Then, each F_{x_k} is calculated as follows:

$$F_{x_k} = tf_{x_k} \cdot log(idf_k) \tag{6.3}$$

where tf_{x_k} is the Term Frequency (TF), computed as the raw frequency of word x_k in a sentence, i.e. the number of times that word x_k occurs in sentence $\mathbf{x}$. idf_k is the Inverse Document Frequency (IDF), which is a measure of how much information word x_k provides, i.e., whether the term is common or rare across corpus P, computed as:

$$idf_k = \frac{|P|}{|\{\mathbf{x} \in P : x_k \in \mathbf{x}\}|} \tag{6.4}$$

$|P|$ is the number of sentences in corpus P, and $|\{\mathbf{x} \in P : x_k \in \mathbf{x}\}|$ is number of sentences of P where word x_k appears.

We will refer to the DDS technique that derives from using TF-IDF in Algorithm 1 as TF-DDS .

6.3.2.2 Continuous vector-space representation

In this thesis, we use other different sentence embedding methods presented in Section 1.4.3. Specifically, in this chapter we use the method proposed by [64], called *Document-vec*. We will refer to this representation by CVR, and to the DDS technique derived from using CVR in Algorithm 5 as CVR-DDS .

6.4 Experiments

In this section, we describe the experimental framework used to assess the performance of the DDS methods described in Sections 6.3.1

and 6.3.2. For this purpose, we studied their behavior in two separated tasks: a controlled scenario with labeled data, and a real e-commerce translation task. We will first describe the experimental setup used, which is common to both tasks, and then we will report on each one of the tasks and their results.

6.4.1 EXPERIMENTAL SETUP

To study to which extent weight optimization could yield improvements in translation quality, and hence obtain an upper bound for the performance of our DDS techniques, we will also report results with the so-called *oracle*, in which tuning was performed directly using the test set. Note that, this setting is not realistic, but is useful to understand how much room is left for improvements by only choosing the development set wisely.

In addition to *oracle*, two more comparative results will be provided: *baseline*, obtained by a translation system where tuning was performed on the original out-of-domain data; and *in-domain*, where tuning was performed using an in-domain development set. They represent a good reference for comparison purposes if we assume that development set is not available.

In this chapter, SMT output will be evaluated using BLEU [80], METEOR [82] and TER [83]. More details about these automatic metrics are given in Section 2.2.

In CVR-DDS (Section 6.3.2.2), two meta-parameters need to be fixed: $size = 200$, the dimension of the vector-space, and $n_c = 1$, the minimum number of times a given word needs to appear in the training data in order to build its corresponding vector. These values were fixed according to preliminary research (see Chapter 3), and kept for all the experiments reported in the current chapter.

6.4.2 CONTROLLED SCENARIO RESULTS

First, we conducted an assessment of our DDS methods (LD-DDS, TF-DDS and CVR-DDS) by analysing their performance in a controlled scenario, where domain labels were readily available. The purpose was to study to which extent the proposed DSS techniques were able

to correctly classify development sentences according to some common feature, for instance domain, by providing a test set belonging to that specific domain.

We resorted to the domain adaptation task from the Johns Hopkins Summer Workshop 2012 [87], where the task was to adapt French-English (FR→EN) models. The training corpus is provided by the parliamentary domain (Canadian Hansards) (corpus details in Section 2.3). Development and test corpora included the medical domain (referred to as EMEA), the general news domain (NEWS), the press domain (PRESS), and the subtitle domain (SUBS). Statistics are provided in Section 2.3.

In this scenario, the development data extracted by our DDS techniques was obtained from a set where all four domain-specific development sets were merged. The *baseline* system was tuned on the Hansards development data, and the *in-domain* system was tuned on the domain-specific development data of each domain, respectively.

6.4.2.1 Precision, Recall and F_1-score

We analysed the ability of our DDS methods to recover the domain labels by providing the corresponding test set. We measured precision, recall and the F_1 scores. The last row, *total*, shows precision, recall and F_1 across all domains in a 4-class confusion matrix (i.e., not the average). Several things should be noted:

- Selecting sentences using CVR-DDS obtained significantly better results than TF-DDS and LD-DDS approaches, except for SUBS, where all methods obtained very similar results.
- The best translation quality was obtained in SUBS domain. We believe that this is because this domain has the largest test corpus, and hence yields better estimations.
- In the case of NEWS, our DDS methods obtained the worst values of precision and recall, which implies that they were not able to retrieve the correct development sentences. This seems to signal that it is not an adequate corpus for adaptation research, as already observed in related work [177, 178].

Table 6.2. Precision, recall and F_1 scores for LD-DDS, TF-DDS and CVR-DDS in the controlled scenario.

		EN-FR			FR-EN		
Domain	System	Precision	Recall	F_1	Precision	Recall	F_1
EMEA	LD-DDS	0.35	0.33	0.34	0.37	0.32	0.34
	TF-DDS	0.16	0.32	0.21	0.16	0.32	0.21
	CVR-DDS	0.64	0.47	0.54	0.74	0.45	0.56
NEWS	LD-DDS	0.10	0.12	0.11	0.08	0.12	0.09
	TF-DDS	0.24	0.28	0.25	0.25	0.60	0.35
	CVR-DDS	**0.16**	**0.53**	**0.25**	0.17	0.54	0.25
PRESS	LD-DDS	0.01	0.01	0.01	0.01	0.02	0.02
	TF-DDS	0.32	0.46	0.38	0.21	0.60	0.31
	CVR-DDS	**0.38**	**0.52**	**0.47**	**0.36**	**0.47**	**0.41**
SUBS	LD-DDS	0.77	0.39	0.51	0.81	0.43	0.56
	TF-DDS	0.74	0.38	0.50	0.38	0.43	0.39
	CVR-DDS	**0.79**	**0.39**	**0.52**	0.74	0.39	0.51
Total	LD-DDS	0.24	0.46	0.31	0.26	0.27	0.27
	TF-DDS	0.35	0.32	0.33	0.24	0.46	0.32
	CVR-DDS	**0.37**	**0.46**	**0.41**	**0.37**	**0.45**	**0.40**

- Finally, the results obtained for the three different methods are coherent across different language pairs (EN-FR and FR-EN).

It is important to note that, the results of LD-DDS depends on threshold τ. In Table 6.2 we only reported the best results obtained, which might slightly bias the results favoring LD-DDS. However, given that LD-DDS is not the best DDS technique (neither in terms of classification metrics, nor in terms of translation quality), we report these results for the sake of assessing its potential.

6.4.2.2 SMT results

Once the quality of the selected development corpus was analysed, we now pursue to establish to which extent classification metrics relate to translation quality. This will be achieved by measuring the

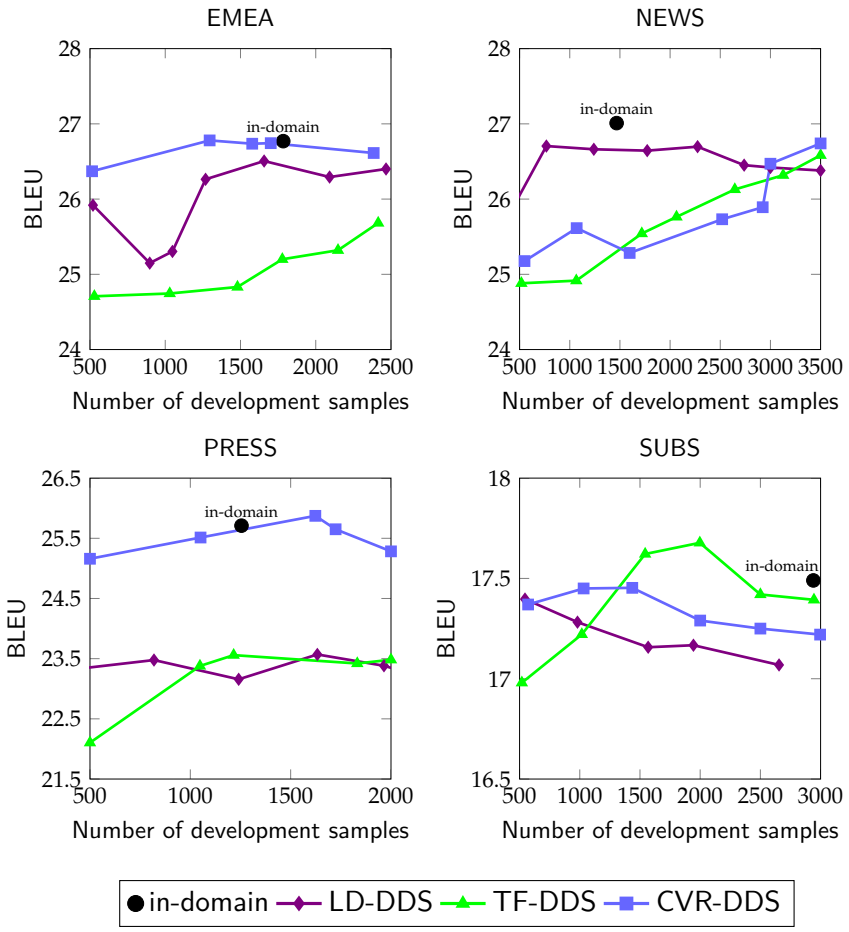


Figure 6.2. Impact caused on BLEU by the variation of the development size in a controlled scenario for EN-FR language pair.

performance of the DDS methods in terms of BLEU. Figures 6.2 and 6.3 show the main results obtained in terms of BLEU with different size of the development corpus obtained with the different DDS. The variation of the development size is calculated changing the radius of the circumference. Several conclusions can be drawn:

- In all domains, DDS techniques are able to yield very similar results to the ones of in-domain baseline. This proves the DDS

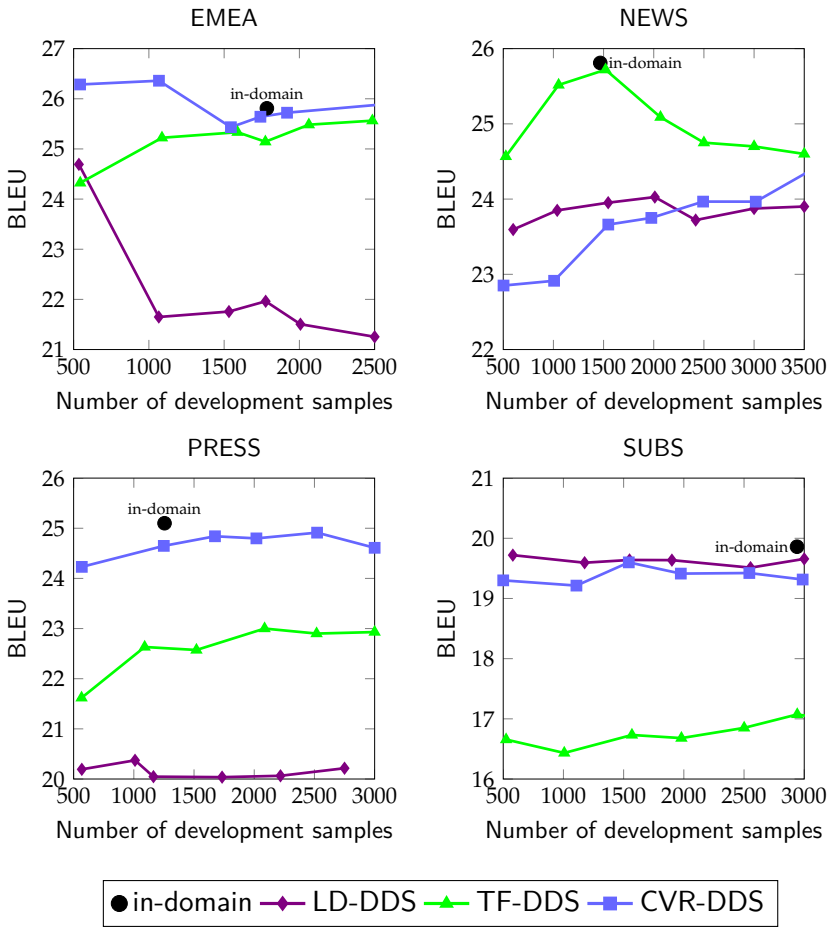


Figure 6.3. Impact caused on BLEU by the variation of the development size in a controlled scenario for FR-EN language pair.

methods effectiveness. It is important to remark that in SUBS domain we are able to achieve the same results with a reduction of the development size.

- CVR-DDS has a more stable behaviour than the other two techniques across different domains and languages.

Table 6.3. Translation results in the controlled scenario. $|S|$ denotes the number of sentences.

		EN-FR				FR-EN			
Domain	System	$\|S\|$	BLEU	METEOR	TER	$\|S\|$	BLEU	METEOR	TER
EMEA	*baseline*	1367	22.1	42.3	74.5	1367	22.7	29.9	62.1
	in-domain	1784	24.8	46.2	68.0	1784	23.8	31.8	60.5
	LD-DDS	1657	24.5	47.7	69.5	535	22.7	30.0	60.0
	TF-DDS	2415	23.7	44.7	68.0	2485	23.6	31.8	62.2
	CVR-DDS	1295	24.8	47.1	67.1	1067	24.4	31.8	61.0
	oracle	1842	26.7	47.5	66.2	1842	26.1	31.9	58.2
NEWS	*baseline*	1367	21.4	41.8	65.2	1367	21.5	29.7	63.6
	in-domain	1467	23.9	46.0	64.7	1467	23.0	30.2	60.1
	LD-DDS	1240	23.7	45.5	63.8	1546	21.0	30.5	67.4
	TF-DDS	3500	23.6	45.3	63.2	1520	23.1	29.8	59.5
	CVR-DDS	3592	23.7	45.4	63.9	3500	22.7	30.3	61.2
	oracle	1782	24.6	46.6	63.9	1255	23.6	30.2	58.9
PRESS	*baseline*	1367	21.9	38.6	66.3	1367	20.5	30.8	62.2
	in-domain	1255	23.9	46.7	64.4	1255	21.1	30.7	58.6
	LD-DDS	1633	21.6	44.0	63.6	2750	17.2	30.1	74.0
	TF-DDS	3500	21.7	44.1	62.5	3500	20.3	31.0	62.5
	CVR-DDS	1724	23.8	46.6	64.2	1674	20.8	30.7	59.6
	oracle	1227	24.6	48.0	64.2	1227	21.8	30.9	58.3
SUBS	*baseline*	1367	16.6	32.6	78.8	1367	12.3	17.3	80.4
	in-domain	2940	18.4	32.7	78.1	2940	18.9	23.6	73.7
	LD-DDS	545	18.4	36.5	79.1	3000	18.7	24.0	75.3
	TF-DDS	1997	18.7	36.3	77.9	3000	15.1	19.7	79.6
	CVR-DDS	1436	18.5	35.4	77.5	1543	18.6	23.2	72.2
	oracle	3281	19.1	36.2	77.2	3281	19.4	24.1	70.1

Table 6.3 shows the best results obtained with different DDS techniques for the test corpus across different domains. We compare the results obtained by DDS techniques with different baseline systems. Several conclusions can be drawn:

- All DDS methods are mostly able to improve over *baseline* across different domains and language pairs. This seems reasonable,

given that the *baseline* results were obtained using an out-of-domain development corpus for tuning purposes.

- CVR-DDS yields better translation quality than LD-DDS and TF-DDS. This seems to signal that CVR-DDS achieves a better representation of the sentences involved. However, results comprising the SUBS domain yield very similar results across all three DDS methods.

- In conclusion, translation quality results between CVR-DDS and *in-domain* are not significantly different. We believe that this is important since it proves the utility of our development DS method, which is able to recover at least a well-suited development set for the task as the development set originally designed for that task.

6.4.3 Real scenario results

After analysing the behaviour of our DDS techniques in a controlled scenario, we pursued to evaluate them in a real-world task, where no development set was readily available. For this purpose, we confronted the system with a set of sentences obtained from a real e-commerce.

We gathered data from e-commence page, *Cachitos de Plata*, details in Section 2.3. Statistics of these corpora are provided in Table 6.4.

As training data, we explored the use of two different corpora available:

1. The United Nations (UN) corpus [94]. Statistics of the corpus provided in Section 2.3.

2. The Common Crawl (COMMON) corpus [93]. Statistics of the corpus provided in Section 2.3.

In this case, our DDS methods were set to sample from the pool of development data available from different years of the WMT task, details in Table 6.4. The *baseline* system was tuned according to the NC development data, all features are provided in Section 2.3.

Table 6.4. Corpora main features of real e-Commerce task. (Dev) is the pool development set.

		e-Commerce		
		$\|S\|$	$\|W\|$	$\|V\|$
EN	Dev	16.4k	330.8k	13.3k
ES			351.5k	15.5k

Given that, no in-domain development set is available, we also considered to randomly sample a set of sentences from the available pool of data, in addition to *baseline* and *oracle*. We will refer to this baseline as *random*. Here, 2500 sentences were randomly sampled from the available pool of development data, without repetition. The results reported show the average of 5 repetitions of the sampling process, where confidence intervals were never greater than 0.2 points (in the corresponding translation quality metric).

Table 6.5 shows the results in terms of BLEU, METEOR and TER, and development set size. Several conclusion can be drawn:

- In all three metrics considered, CVR-DDS achieves consistent improvements over the *baseline* translation quality.
- In all three metrics, CVR-DDS achieves consistent improvements over the *random* translation quality, across both language pairs with fewer sentences. Note that, it is typically assumed that such random baseline is very tough to beat in DS and active learning research [159,160]. Furthermore, improvements are statistically significant.
- Training with UN and COMMON leads to very different results. We assume this happens because although COMMON is a smaller corpus, it is more related to the domain at hand. Common crawl data is crawled from the web, and in this case we are dealing with web data.

Table 6.5. Translation results of real e-commerce scenario.

		EN-ES				ES-EN			
Training	System	$\|S\|$	BLEU	METEOR	TER	$\|S\|$	BLEU	METEOR	TER
UN	*baseline*	2600	13.8	42.2	67.3	2600	17.4	24.9	60.8
	random	2500	12.5	40.9	64.7	2500	18.2	27.4	60.9
	LD-DDS	1657	11.4	39.4	64.6	2334	17.1	27.0	61.2
	TF-DDS	2418	13.1	38.6	67.6	**2610**	**19.2**	**27.9**	**59.9**
	CVR-DDS	**1681**	**14.8**	**42.8**	**64.4**	1750	18.6	27.9	60.2
	oracle	886	19.3	45.6	58.3	886	21.0	28.8	58.0
COMMON	*baseline*	2600	20.2	49.9	55.0	2600	24.1	32.9	52.7
	random	2500	21.9	49.7	56.6	2500	22.1	33.1	52.2
	LD-DDS	2452	20.1	49.9	54.9	2515	21.5	33.5	60.6
	TF-DDS	2628	22.1	50.4	53.8	2580	24.8	34.4	52.5
	CVR-DDS	**2346**	**22.8**	**50.0**	**56.6**	**2445**	**25.6**	**34.7**	**51.4**
	oracle	886	31.1	55.7	53.3	886	33.0	37.4	43.5

6.5 Summary

In this chapter, we have presented different techniques for building a test-specific development corpus, leveraged for optimizing the log-linear weights of the SMT system. We proposed three new DDS methods: LD-DDS, TF-DDS, and CVR-DDS. We analysed the performance of these methods in a controlled scenario, where domain labels are available, and evaluated the methods in a real translation task using e-commerce data without a development set readily available. The empirical results show that CVR-DDS, leveraging a continuous vector-space representation of the sentences, is able to improve over baseline translation quality; and provides a development set that leads to a similar translation quality obtained whenever an in-domain development set is readily available. In addition, the results obtained with CVR-DDS consistently and significantly improve over those obtained with a random sampling baseline across different languages.

7 Data selection in NMT

* * *

"Mr. and Mrs. Dursley, of number four Privet Drive, were proud to say that they were perfectly normal, thank you very much."

—J.K. ROWLING
HARRY POTTER AND THE PHILOSOPHER'S STONE

"El señor y la señora Dursley, del número cuatro de Privet Drive, estaban orgullosos de decir que eran perfectamente normales, muchas gracias."

—FREETRANSLATION SDL TRANSLATOR
HARRY POTTER Y LA PIEDRA FILOSOFAL

* * *

7.1 Introduction

During the last years, a major development in SMT has been the use of neural networks. One of the main advantages of NMT is a better sharing of statistical evidence between similar words and inclusion of rich context [79].

Motivated by the success of Data Selection in PBSMT, we investigate in this chapter to what extent and how NMT can benefit from DS as well. DS has been applied to NMT to reduce the size of the training data [65, 179]. In addition, other works confirmed that NMT systems are known to under-perform when trained on limited parallel data [180], so this is a challenging task. To mitigate the negative effect caused by training a NMT system with a small amount of parallel data, different alternatives have been proposed [130, 181, 182].

Typically, in these works the selected corpus is used during the fine-tuning process in diverse ways.

In this chapter, we introduce the use of the different DS techniques (proposed and used in PBSMT, see Chapters 3, 4, and 5) in the context of NMT, and explain how they are applied to adapt an NMT system. The chapter is divided in two parts:

1. The first part focuses on comparing the effect caused by the use of a common data selection approach (increasing the training corpus) in PBSMT and NMT (Section 7.2).

2. The second part focuses on the use of synthetic data in NMT (Section 7.3). Different works demonstrated that the combination of real parallel corpora with synthetic bilingual data enhances the NMT translation quality ([110]). In this part, we propose to use DS techniques to build an appropriate set of synthetic data.

The rest of the chapter is organized as follows: Section 7.2 reviews the use of DS techniques to select bilingual data using a NMT system. Section 7.3 presents the DS methods used to construct synthetic data, which will be further used to adapt a NMT system. Experiments are presented in Section 7.4, and the conclusions drawn from results are presented in Section 7.5.

Table 7.1 shows the abbreviations introduced in the current chapter, in order to facilitate a better comprehension of the text.

Table 7.1. Abbreviations used in Chapter 7.

Abbreviation	Description
SMT	Statistical Machine Translation
PBSMT	phrase-based SMT
NMT	Neural Machine Translation
DS	Data Selection
CVR	Continuous Vector-space Representation
CE	Cross-Entropy method
CRSDS	Continuous Vector-Space Representation of Sentences for DS
NNCDS	Neural Network Classifier of Sentences for DS

7.2 DS for training PBSMT and NMT approaches

Data selection has an important role in the PBSMT paradigm. In this section, we present a comparison of the effect of using DS within these two paradigms. The idea is to know if the benefit of selecting the best corpus of training is also extensible for NMT. The training corpus was obtained using the same criteria as detailed in Chapter 3. The training corpus is the concatenation of the corpus in the domain and the selected corpus, captured by some DS methods. DS methods used for comparison purposes are the same as the ones presented in Chapter 3:

- Cross-Entropy method (CE), described in detail in Section 3.4.1.
- Continuous Vector-Space Representation of Sentences for Data Selection (CRSDS), described in detail in Section 4.2.
- Neural Network Classifier for Data Selection (NNCDS), described in detail in Section 4.3

7.3 Data selection to create synthetic data

Synthetic parallel data have been widely used to boost the translation quality of NMT. In this section, we propose a new method for adapting a general NMT system to a specific task (source test corpus only), by exploiting synthetic data.

In certain language pairs or domains where parallel corpora are scarce or even non-existent, a model adjusted with synthetic data can improve the performance regarding a more general model [110]. Once a model has been trained on a large, general corpus, we can adapt it to a new domain by fine-tuning it exclusively using the synthetic data. To accomplish this, we create an ad-hoc, specific synthetic corpus in which the features from our target-domain data are present. This corpus is built by selecting those instances that are related with our source test set from a large monolingual pool of sentences (in the source language). Next, we automatically translate these sentences into the target language. Finally, using this synthetic corpus, we fine-tune a NMT system trained on a more general domain. Figure 7.1

shows the pipeline of our adaptation process. In the next section, we describe our technique for creating adequate synthetic corpora and the NMT adaptation process.

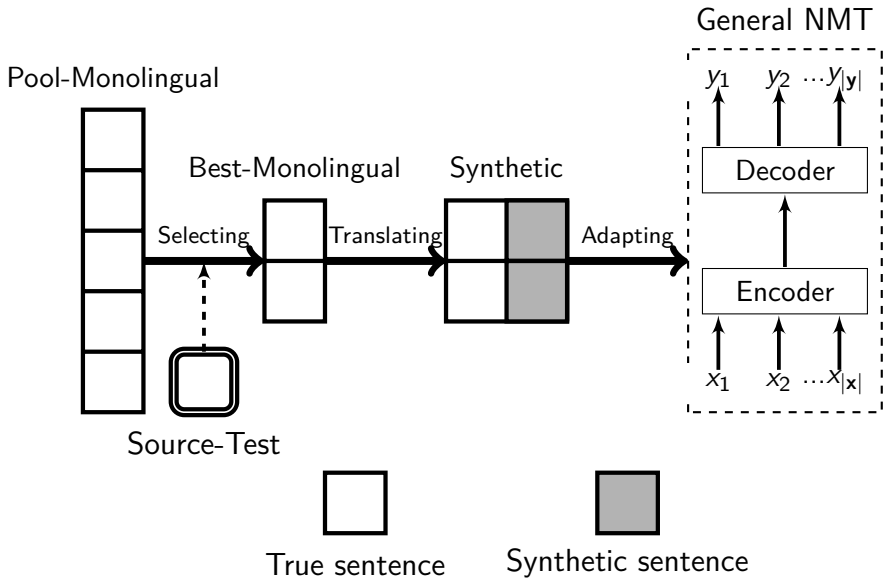


Figure 7.1. Adequate synthetic parallel corpus building process for a given test set.

7.3.1 Synthetic data creation method

For creating an adequate synthetic corpus for adapting an NMT system, we select from a large pool of monolingual text the most related sentences to our task at hand. The DDS method proposed in Chapter 5 can be used for this task. More specifically, DDS with continuous vector space representation will be used. Given that this method was already explained in Section 6.3.2, the reader is refereed to that chapter for further details. To sum up, the selection algorithm performs as follows:

1. Represent all sentences (source test and source monolingual corpora) in a common continuous space.

2. Construct a hyper-sphere that contains the source test data.

3. Select all monolingual out-of-domain samples that are inside the hypersphere.

7.4 Experiments

In this section, the experimental setup is described. Section 7.4.2 presents the experiments performed regarding different ways to leverage a DS method to increase the translation quality of a NMT system. Finally, in Section 7.4.3, we present the results obtained using synthetic data to adapt a NMT system to a specific domain.

7.4.1 Experimental setup

We used NMT-Keras [98] for building the NMT system, as described in Section 1.5. We applied joint byte pair encoding (BPE) [183], learning $32k$ merge operations. Following the findings from [184], we used LSTM units. Due to practical reasons, we used single-layered LSTMs. The LSTM, word embeddings and attention MLP sizes were 512 each. We applied layer normalization [185] and Gaussian noise ($\sigma = 0.01$) to the weights [186]. We clipped the L_2 norm of the gradients to 1 [187]. We used the Adam optimiser [157] with a learning rate of 0.0002 [188]. The size of the beam was set to 6.

7.4.2 Training a NMT system

We first compare the effects of a commonly used DS method on both NMT and PBSMT. Concretely, we used three different DS methods: Cross-Entropy method (CE), Continuous Vector-Space Representation of Sentences for Data Selection (CRSDS) and Neural Network Classifier of Sentences for Data Selection (NNCDS). The section was divided into two parts: the first part introduces the corpora employed and second part presents the experimental results and discussion.

7.4.2.1 Corpora

We evaluated all experiments on the IT domain (details in Section 2.3) across language pairs directions (EN→CS and CS→EN). The out-of-domain corpus used was the Europarl corpus, details in Section 2.3.

7.4.2.2 *Experiments and discussion*

In this section, the experimental results are presented by comparing NMT and PBSMT using various DS methods. Figure 7.2 shows the translation performance in terms of BLEU for the development corpus on the IT domain using two language pairs. Some conclusions can be drawn:

- The benefits of the DS methods for PBSMT are confirmed. In all test sets, the selection method yields better performance than using only the in-domain data (green hexagon). Selection methods using only 10% of the out-of-domain corpus provide comparable results to the use of all available data (light green line). We also show that the informed selections are superior to random selections of the same size (dashed green line).
- In NMT, the results of the data selection methods are different. Interestingly, for systems with similar sizes to those proven to be useful in PBSMT, the DS-adapted NMT system is not able to beat the system built with all available data (light violet line), indicating that NMT suffers much more from small-data setting compared to PBSMT, even if the training corpus is more adequate for the translation of the specific domain. Besides, the random selection (dashed violet line) shows that NMT not only needs large quantities of data, but it is also affected when the selected data exhibits low quality.

Table 7.2 provides a translation result summary (test corpus result) of the different systems trained using the two paradigms (phrased-based SMT and NMT). The results confirmed the conclusions exposed before: the same selection size corpus used to train a SMT system is more beneficial in the case of PBSMT than in the case of NMT. We can conclude that DS methods that are an appropriate choice for PBSMT are not the most adequate option for NMT.

7.4.3 FINE TUNING WITH SYNTHETIC DATA

In this section, we describe the experimental framework employed to assess the performance of the NMT adaptation method described

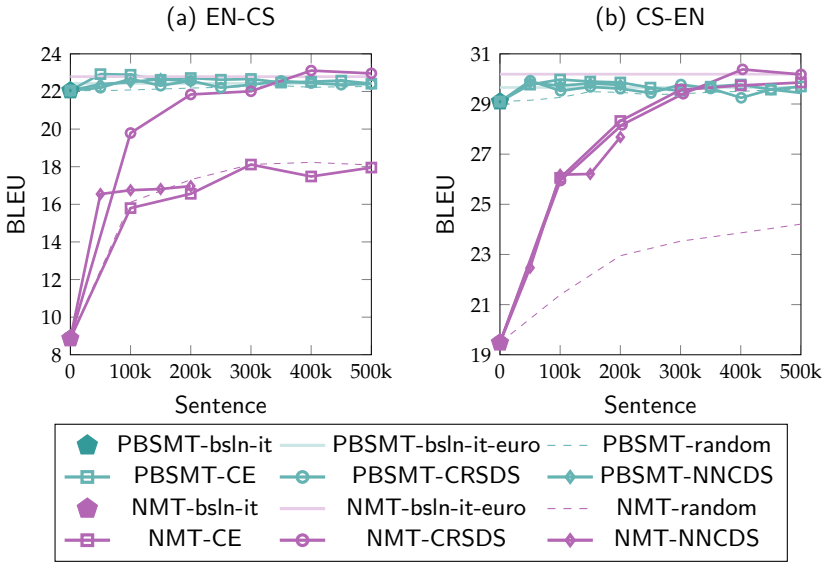


Figure 7.2. Impact caused on BLEU metric by the addition of sentences using CRSDS, NNCDS, CE, and random monolingual DS in the Medical domain. Horizontal lines represent baseline systems (bsln-it-euro), PBSMT represents the system using phrase-based SMT and NMT represents the system using neural networks.

in Section 7.3. For this purpose, we study its behaviour using three corpora.

7.4.3.1 Corpora

We performed the experiments on English-Spanish (EN→ES) translation. Our out-of-domain training data was the Common Crawl (COMMON) corpus [93], details of this corpus are given in Section 2.3. We chose the One Billion Word corpus [95] as the large pool of monolingual sentences, details of this corpus are given in Section 2.3. We chose the News-Commentary (NC) test dataset for validation, details of this corpus appear in Section 2.3.

We used corpora from three different domains for testing:

1. XRCE printer manuals (XRCE–Test), details in Section 2.3.
2. Information Technology (IT–Test), details in Section 2.3.

Table 7.2. Translation comparison between using NMT and PBSMT for DS methods. Columns denote, from left to right: SMT paradigm, DS methods, $|S|$ for number of sentences which are given in terms of the in-domain corpus size, and $(+)$ the number of sentences selected, BLEU. CE represent Cross-Entropy method, CRSDS for Continuous Vector-Space Representation of Sentences for Data Selection method and NNCDS for Neural Network Classifier for Data Selection method.

		EN-CS				CS-EN			
	System	$\|S\|$	BLEU	METEOR	TER	$\|S\|$	BLEU	METEOR	TER
PBSMT	bsln-it	123k	15.9	23.8	61.5	123k	22.5	31.9	55.8
	bsln-it-euro	123k+536k	16.8	24.7	59.2	123k+536k	23.4	32.7	55.8
	Random	123k+350k	16.3	24.1	60.7	123k+500k	23.6	32.9	53.7
	CE	123k+50k	16.8	24.7	59.7	123k+100k	23.5	32.4	59.7
	CRSDS	**123k+100k**	**17.1**	**24.9**	**60.2**	**123k+50k**	**23.7**	**32.8**	**54.3**
	NNCDS	123k+200k	16.6	24.7	60.6	123k+50k	23.6	32.8	54.3
NMT	bsln-it	123k	6.19	10.7	75.1	123k	16.5	19.8	65.6
	bsln-it-euro	**123k+536k**	**19.0**	**24.4**	**58.3**	**123k+536k**	**25.8**	**31.4**	**53.4**
	Random	123k+400k	15.2	21.4	61.6	123k+500k	20.1	27.5	62.0
	CE	123k+300k	16.1	21.4	64.1	123k+500k	25.3	30.8	54.4
	CRSDS	123k+400k	18.7	24.3	58.2	123k+400k	25.1	30.9	54.3
	NNCDS	123k+200k	16.5	21.8	61.1	123k+200k	24.1	30.2	55.4

3. Electronic Commerce (e-Commerce–Test). This last corpus was obtained from a real e-commerce website (*Cachitos de Plata*). We introduced the use of this corpus in Chapter 6. All features are given in Section 2.3.

7.4.3.2 Corpus creation

The process for building synthetic parallel corpora begins with the selection of data from monolingual pool. The selection method presented in Section 7.3.1 requires to set the dimension of the vector-space representation. We set it to $size = 200$ according to what is described in previous Chapters. Once, we obtained the monolingual selections, they are translated. In order to speed up this process, we split the selection and translated in parallel using Moses and NMT. Both systems were trained on the out-of-domain data, in this case COMMON corpus.

7.4.3.3 *Analysis of the selection*

Table 7.3 shows some features of each corpus selected. Note that, the average length of the sentences belonging to each selection is tightly related to the sentence length of each test set.

Table 7.3. Features of the monolingual selections for each test set (T), employed for adapting the NMT system. $|S|$ denotes the number of sentences; $|W|$ is the number of words; $|V|$ is the vocabulary size and $\overline{|W|}$, the average sentence length.

T		$\lvert S\rvert$	$\lvert W\rvert$	$\lvert V\rvert$	$\overline{\lvert W\rvert}$
XRCE – Test	EN	180k	2.2M	54k	9.4
	ES		1.7M	58k	12.2
IT – Test	EN	150k	2.5M	76k	16.7
	ES		3.0M	78k	20.0
e-Commerce – Test	EN	300k	3.2M	100k	10.6
	ES		4.1M	100k	13.6

Therefore, the selections of XRCE and e-Commerce had shorter sentences, while the selection obtained from the IT corpus had longer sentences. As shown in the following sections, this was a key factor that affected the performance of the machine translation systems.

Moreover, Table 7.4 shows some samples of each domain, selected by our selection technique. We can notice that, such samples are related to the corresponding test set domain. Thus, sentences from XRCE and IT domains refer to a technological field. As illustrated in Table 7.3, sentences selected from the IT corpus were notoriously longer than those selected from XRCE. Sentences selected from the e-Commerce task are related to jewelry or economy. Given the e-Commerce domain (an electronic shop of silver jewelry), the sentences obtained are also coherent.

7.4.3.4 *Translation results*

Table 7.5 shows the results obtained for XRCE and IT tasks. In NMT experiments, we show results using 4-model ensemble. Some conclusions can be drawn:

Table 7.4. Selection examples from each domain.

Domain	Selected sentence
XRCE	• id rather send files electronically • use current antivirus and a firewall • images are stored on a one terabyte built in hard drive which includes a DVD burner
IT	• the technology would also be available to ipod touch users although they would have to buy a microphone and headphones to make calls pc world reported • if you want to find panorama archive material on delicious the easiest way to search is to use the single word on the right hand column • my personal have is tweetdeck which although designed for photo uploading amongst other things
E-Com	• it is perfect for your collection • pasta is inexpensive easy and really romantic • another shows the dust forming into clumps along magnetic lines like pearls on a necklace

- The general NMT model performed worse than Moses in out-of-domain tasks. General NMT model with 4-model ensemble was very helpful (Σ), nevertheless, it still has a lower performance than Moses.

- In XRCE task, TER values of the general NMT system were unusually high. This unexpected results may be due to the some corpus features: the XRCE corpus has an average sentence length of 9 words while the general NMT model generated sentences with an average of 13 words (it was trained on general-domain data). The TER metric greatly penalizes this behaviour, because of it must delete the exceeding words yielding surprisingly higher values. In the case of Moses, the average sentence length generated was 9.5. The reason for this is that the generation was bounded by the phrase and language models.

- The addition of synthetic data significantly improved the NMT systems in all cases. Taking as reference a single NMT model, the gains ranged from 5 to 7 BLEU points. The performance of a single fine-tuned NMT model was also clearly better than fine-tuned ensembles.

- Especially critical were the enhancements in terms of TER metric. In XRCE task, the synthetic data improved by almost 40 and 20 TER points, for single and ensemble models, respectively. Due to the addition of synthetic data, the system learned produces shorter translations (around 5 words shorter, on average), greatly diminishing TER values. In IT task, the synthetic data also improved TER, but to a lower extent. This is because of the IT task is closer to the out-of-domain corpus. Therefore, the adaptation benefits brought by the synthetic data were less crucial tin IT task than in XRCE task.

Table 7.5. NMT adaptation translation results for XRCE and IT tasks. BLEU and TER results are given in percentage. $^{\Sigma}$ denotes an ensemble of 4 neural models. $\overline{|W|}$ is the average number of words per sentence.

	XRCE			IT		
System	BLEU	TER	$\overline{\|W\|}$	BLEU	TER	$\overline{\|W\|}$
Moses	26.2 ± 0.8	59.0 ± 0.8	9.1	33.4 ± 0.6	45.6 ± 0.6	20.4
NMT	20.4 ± 1	94.5 ± 5.1	12.8	29.0 ± 0.8	53.5 ± 0.8	15.3
NMT$^{\Sigma}$	25.5 ± 0.8	76.8 ± 2.0	11.3	31.4 ± 0.8	51.2 ± 0.8	15.3
NMT + Synthetic	27.5 ± 0.8	56.7 ± 0.8	8.6	34.1 ± 0.7	45.7 ± 0.7	17.8
NMT$^{\Sigma}$ + Synthetic	27.3 ± 0.8	56.3 ± 0.8	8.4	33.8 ± 0.7	46.3 ± 0.7	18.1

Table 7.6 shows the results on real E-Com task. This was a very specific task and can be concluded that:

- The general NMT model also yielded worse performance in terms of BLEU than Moses.However, when applying a model ensemble, the results were significantly enhanced. In terms of BLEU, it even managed to beat Moses.

- NMT systems behaved similarly, in terms of TER, as in the case of XRCE task . The E-Com corpus has similar features to XRCE–Test (in this case, 9.7 words per sentence). Therefore, the same phenomenon was observed: as we introduced in-domain-related sentences, the system learned produces shorter sentences, consequently diminishing TER.

- The use of synthetic data again greatly improved the system. The results were in agreement with previous experiments: a single fine-tuned model, significantly outperformed the general system (+9 BLEU points). A sole adapted system was even better than a general model ensemble. With respect to Moses, we also found major enhancements in terms of BLEU.

- It is also noticeable that the ensemble of systems trained with synthetic data did not improve the performance of a single fine-tuned system. This is probably due to the fact that the adaptation was performed from an already trained model and with fewer data. The systems belonging to the ensemble were quite similar, all of them around the same local minimum. Therefore, potential enhancements derived from the ensembles were diluted.

- Finally, we should point out that the E-Com task belongs to a real-world scenario. This corpus is not designed for experimental purposes. It contains elements that distort the experiment, and therefore lead to unpredictable results. An excellent example of this problem is the Spanish phrase: "plata de ley". Translation systems are not able to find the correct translation and the best hypothesis is "plata esterlina". In this case, the automatic metrics are unable to measure the translation quality correctly. Hence, in such open scenarios, a human evaluation should be the next step to take [189].

7.4.3.5 Translation examples

Translation instances from each corpus are shown in Table 7.7. First, let us define some notations for this qualitative results which will be useful for the analysis: Src denotes the source sentence to translate, Moses denote the output obtained by the Moses system, NMT represents hypothesis obtained by NMT paradigm, NMT^{Σ} is the NMT system with ensemble, NMT + Synth is the output from the NMT adapted using synthetic corpus and Ref denotes the sentence reference.

Table 7.6. E-Commerce – Test set results. BLEU and TER results are given in percentage. $^{\Sigma}$ denotes an ensemble of 4 neural models. $\overline{|W|}$ is the average number of words per sentence.

	E-Com		
System	BLEU	TER	$\overline{\|W\|}$
Moses	21.1 ± 0.8	56.7 ± 0.7	9.4
NMT	16.9 ± 1.0	84.6 ± 6.3	14.1
NMT$^{\Sigma}$	23.0 ± 1.0	75.6 ± 2.9	12.0
NMT + Synthetic	25.5 ± 1.0	59.1 ± 1.0	8.7
NMT$^{\Sigma}$ + Synthetic	25.8 ± 1.0	60.8 ± 2.6	8.7

In the first example for XRCE corpus, all the systems presented the same error at the beginning of the translation (*especificación del*). This is because it was the most likely translation in our corpus, both in the real and synthetic ones.

In the second example, Moses was not able to correctly identify the right meaning of the word (*windows*) in the sentence to translate. It should be left untranslated, as it is a proper noun. The NMT systems were able to detect it. Also, Moses, NMT and NMT+Synth systems presented the same lexical choice error at the word (*deberían*).

Lastly, we show translation examples for the e-commerce domain. Moses obtained the worst translation. The NMT$^{\Sigma}$ method was not able to produce the correct translation (*precioso*), as provided in the reference, but instead produced a synonym (*hermoso*). It is important to consider that, although this may not be a real mistake in translation terms, it will be penalized by BLEU and TER metrics. The NMT+Synth provided the closest translation to the reference. Even though, the system was unable to obtain a translation for the word (*intertwined*).

7.5 Summary

In this chapter, the use of DS methods for NMT paradigm has been introduced. The chapter was divided in two main parts: On the one hand, we focused on training a NMT system using a set obtained

Table 7.7. Translation examples using MT systems built for each domain: Src (source sentence), Moses (moses system), NMT (NMT system), NMT$^{\Sigma}$ (NMT system with ensemble), NMT + Synth (NMT using synthetic corpus) and Ref (reference).

XRCE	Src	• specifying the output file format 2-29
	Moses	• *especificando el* formato de salida 2-29
	NMT	• *especificar el* formato de archivo de salida 2-29 .
	NMT$^{\Sigma}$	• *especificar el* formato de archivo de salida *de 29 a 29* .
	NMT+Synth	• *especificar el* formato de archivo de salida 2-29
	Ref	• especificación del formato de archivo de salida 2-29
IT	Src	• almost all apps installed on windows 8 should work correctly in windows 8.1
	Moses	• casi todas las aplicaciones instaladas en *las ventanas* 8 *debería* funcionar correctamente en *ventanas* 8.1
	NMT	• casi todas las aplicaciones instaladas en windows 8 *deben* funcionar correctamente en windows 8.1
	NMT$^{\Sigma}$	• casi todas las aplicaciones instaladas en windows 8 deberían funcionar correctamente en windows 8.1
	NMT+Synth	• casi todas las aplicaciones instaladas en windows 8 *debería* funcionar correctamente en windows 8.1
	Ref	casi todas las aplicaciones instaladas en windows 8 deberían funcionar correctamente en windows 8.1
E-Com	Src	• they are a lovely set of small and thin strips silver intertwined
	Moses	• son un *conjunto* de pequeñas y *encantadoras tiras finas plata interrelacionado*
	NMT	• son un precioso conjunto de *tiras de película pequeña y delgada*
	NMT$^{\Sigma}$	• son un *hermoso* conjunto de pequeñas y finas tiras de plata
	NMT+Synth	• son un precioso conjunto de pequeñas y finas tiras de plata
	Ref	• son un precioso conjunto de pequeñas y finas tiras de plata entrelazada

by DS method, concatenating the in-domain corpus and the selection corpus. A comparison with the PBSMT paradigm was presented in the experiments section across different language directions. Results show that is necessary to look for other alternatives to deal with the selected corpus.

On the other hand, we present the use of a DS method to create a synthetic corpus. The synthetic corpus is employed to fine-tuning a NMT system. This adaptation method has been applied to three domains. Results show significant improvements in terms of BLEU with respect to the original model. It is also worth mentioning that,

once the selection was performed, the adaptation of NMT systems to new domains was very fast.

8 Conclusions

* * *

"La Julieta va venir expressament a la pastisseria a dir-me que, abans de rifar la toia, rifarien cafeteres; que ella ja les havia vistes: precioses, blanques, amb una taronja pintada, partida en dues meitats, que ensenyava els pinyols."

—MERCÈ RODOREDA
LA PLAÇA DEL DIAMANT

"Julieta came on purpose to the confectionery to say me that, before rifar the toia, rifarien cafeteres; that she already had seen them: lovely, white, with an orange painted, split in two halves, that taught the stones."

—APERTIUM TRANSLATOR
THE SQUARE OF THE DIAMOND

* * *

8.1 Summary

It is especially obvious the need to bring effective SMT systems to many practical uses, for instance, automatically translation online, interacting with diverse customers over instant messaging or broadening e-Commerce web presence in new markets. In all these SMT different scenarios, the corpora is still a gating factor for translation quality. This thesis supports the hypothesis that it is necessary to find out relevant or adequate data to build or adapt machine translation systems.

In Chapter 4, we focused on the selection of the best training corpus for PBSMT systems. We presented innovative DS techniques for

selecting the adequate training corpus for a specific domain. The intuition behind these techniques is based on the continuous vector representation of the words or sentences. In statistical machine translation, experimental results show that selection methods produce a significant increase on the translation quality together with a reduction of the training corpus size. The experiments were carried out using state-of-the-art PBSMT systems, covering several different language pairs, and with corpora used in standard machine translation tasks.

The training set selected obtained by a DS method could be used in different ways than used in Chapter 4. In Chapter 5 we provided different options to make other use of the selected training corpus. The experimental results were demonstrated the viability of DS methods in PBSMT.

Tuning process is an essential step in PBSMT, and the corpora employed during this process have a significant impact on the adaptation process. In Chapter 6, we focused on the creation of the best development corpus for those cases where only a specific test set was available. To solve this problem, we have used three DDS methods. To prove the viability, we analyse the results in two different scenarios. The results obtained from such scenarios point towards a potential benefit which can be achieved by applying the techniques described.

Over the past years, NMT paradigm made progress in MT state-of-the-art methods and corpora continue to be relevant over the translation quality. For this reason, we dedicated Chapter 7 its study. On the one hand, we suggest to use DS methods (proposed in Chapter 3) in NMT paradigm. The results obtained on different domains are promising and evidence the potential benefit that can be achieved by applying the techniques described. On the other hand, we developed a strategy for increasing the adaptability of a NMT system using synthetic corpus. This strategy is inspired by the idea of employing a DS method to construct a synthetic bilingual corpus. DS method aims to select the most relevant part of a monolingual corpus. The improvements obtained in this task are positive and are in agreement with all the experiments performed, involving different corpora.

All in all, the main contributions of this thesis are:

1. Selecting the adequate data can yield a great benefit for PBSMT. We present different DS techniques to increase the size of the in-domain training corpus. Results reported on different domains and language pairs provide consistent improvements and reduction of training size.

2. DS methods show their benefit in the creation of the development corpus. Results reported on a real scenario are very positive.

3. We studied the use of DS methods in NMT paradigm. Some techniques were applied to PBSMT in same way in this paradigm.

4. A DS method was used to create a synthetic corpus. The synthetic corpus was used to perform a fine-tuning process. Experimental results showed translation improvements. These results demonstrate the benefits of using DS methods in NMT.

8.2 Future works

Research is a never-ending field of work and although this thesis is completed, a large amount of work remains to be done.

As mentioned previously, NMT approaches are taking off nowadays. They have broken the performance of PBSMT and have shown a promising performance. Continue applying the DS ideas proposed in current NMT approaches could be one of the directions to follow for future research work. This thesis may act as a useful reference for the SMT community, specially for people working or that will work on NMT adaptation approaches. The first direction we intend to explore is the use of monolingual corpus. It would be interesting to analyse different alternatives to introduce bigger monolingual corpora in NMT models and to study the creation of synthetic corpus specifically, in languages where only a little corpus is available. As a second direction, we propose to investigate the combination of models.

A Log-linear weight adaptation

* * *

"El dolor es inevitable en el paso por esta vida, pero dicen que casi siempre es tolerable si no se le opone resistencia y no se agregan miedo y angustia."

—ISABEL ALLENDE
PAULA

"The pain is inevitable in the passage by this life, but they say that almost always it is tolerable if resistance is not against to him and they do not add fear and distresses."

—SYSTRANET TRANSLATOR
PAULA

* * *

A.1 Introduction

The aim of adaptation techniques is to make the most effective use of a small amount of adaptation data (belonging to the domain translated afterwards); in order to take advantage of the generality provided by the massive amount of data available in more resourceful domains. A popular approach is to adjust the log-linear weights (λ) present in PBSMT systems. The objective is to improve the performance of the system by optimizing these weights in the target domain.

In this chapter, we will be focusing on adapting each of the log-linear weights (λ) in Equation 1.5. Appropriate values of such parameters for a given domain do not necessarily imply a good combination in other domains. Besides, we introduce the process for obtaining the

best scaling factors λ; called tuning. The most popular tunning algorithms are: Minimum Error Rate Training (MERT) [17] and batch Margin Infused Relaxed Algorithm (MIRA) [23]. These methods were already introduced in Section 1.3.1.1. For this task, an optimisation algorithm based on the Discriminative Ridge Regression (DRR) technique is presented and compared the two state-of-the-art methods.

The main contributions of this chapter are:

- We present an algorithmic description of DRR in both variants, as applied for the estimation of the log-linear weights in an adaptation scenario.
- We empirically evaluate the DRR algorithm proposed in three different domains using different language pairs.
- We provide a thorough comparison with state-of-the-art λ estimation methods, such as MERT and MIRA.

This chapter is structured as follows: In Section A.2, we describe the algorithmic approach for applying DDR to estimate λ in an adaptation scenario. In Section A.3, the experimental design and empirical results are detailed. Finally, conclusions are explained in Section A.4.

Table A.1 shows the abbreviations introduced in the current chapter, in order to facilitate a better comprehension of the text.

Table A.1. Abbreviations used in Chapter 8.

Abbreviation	Description
SMT	Statistical Machine Translation
DRR	Discriminative Ridge Regression
MERT	Minimum Error Rate Training
MIRA	Margin Infused Relaxed Algorithm

A.2 Discriminative ridge regression for SMT

In this section, the Discriminative Ridge Regression (DRR) method for estimating λ is introduced. DRR, as proposed by [190], uses the concept of ridge regression to develop a discriminative algorithm in

order to estimate $\boldsymbol{\lambda}$ on-line, i.e., as new adaptation samples are introduced into the system.

The key idea is to find out a configuration of the weight vector using all hypotheses within a given n-best list. In this way, good hypothesis are rewarded and bad hypothesis are penalised trying to narrow the correlation between the score function σ , and the quality criterion used (BLEU or TER metric).

Since DRR was proposed for an on-line computer-assisted translation scenario, it requires an n-best list of hypotheses for each of the sentences that are evaluated by the professional translator post-editing the system's output.

In this chapter, we propose two different variations of DRR for optimising the log-linear weights. The first option is called, sentence-by-sentence DRR. In this case, $\boldsymbol{\lambda}$ is obtained by adjusting the vector after observing each sentence in a development corpus. The second alternative is batch DRR, where the optimisation process is performed by using a batch for development sentence.

A.2.1 SENTENCE-BY-SENTENCE DRR

In this section, we present sentence-by-sentence DRR. Algorithm 6 shows the procedure. Here, we have a bilingual development corpus $\mathcal{A}$, where $\mathcal{F}$ is the number of sentences in the development corpus $\mathcal{A}$ ($\mathcal{F} = |\mathcal{A}|$), $f \in \{1 \dots \mathcal{F}\}$, and $\mathcal{I}$ is the maximum number of epochs desired.

During the **optimization** step in Algorithm 6, we obtain the vector $\check{\boldsymbol{\lambda}}$ for each of the development sentences $\mathbf{x}_f$. Within DRR, this optimisation is performed by computing the solution to an overdetermined system (described in detail in next section) so that changes in the scoring function σ are correlated to changes in the objective function (potentially some automatic evaluation metric like BLEU or TER).

A.2.1.1 Sentence-based optimisation in DRR

As exposed in the previous section, DRR obtains $\boldsymbol{\lambda}$ by computing the best vector for each of the sentences in the development corpus. In

Data: Development corpus $\mathcal{A}$
Result: $\boldsymbol{\lambda}$
Initialize: $\boldsymbol{\lambda}^0$;
forall *desired number of iterations* $\mathcal{I}$ **do**
  forall *number sentences in dev-corpus* $\mathcal{F} = |A|$ **do**
    optimization: compute vector $\check{\boldsymbol{\lambda}}_i^f$;
    estimation: $\boldsymbol{\lambda}_i^f = (1-\alpha)\boldsymbol{\lambda}_i^{f-1} + \alpha\check{\boldsymbol{\lambda}}_i^f$;
  end
end
selection: output vector $\boldsymbol{\lambda}_{\mathcal{I}}^f$

Algorithm 6: Pseudo-code for DRR estimating $\boldsymbol{\lambda}$ as described in Section A.2

order to compute the new log-linear weight vector $\boldsymbol{\lambda}^f$, the previously learned $\boldsymbol{\lambda}^{f-1}$ needs to be combined with an appropriate update step $\check{\boldsymbol{\lambda}}^f$. The aim is to compute an appropriate update term $\check{\boldsymbol{\lambda}}^f$ that better fits the translation search space (approximated to a n-best list) of the development sentence pair observed at f. This is often done as a linear combination [191], where:

$$\boldsymbol{\lambda}^f = (1-\alpha)\boldsymbol{\lambda}^{f-1} + \alpha\check{\boldsymbol{\lambda}}^f \tag{A.1}$$

for a certain learning rate α.

Let n-best($\mathbf{x}$) be such a list computed by our models for sentence $\mathbf{x}$. To obtain $\check{\boldsymbol{\lambda}}^f$, we define a $N \times M$ matrix $\mathcal{H}_{\mathbf{x}_f}$ that contains the feature functions $\mathbf{h}$ of every hypothesis. M is the number of features in Equation 1.5, and N is the size of n-best($\mathbf{x}$).

$$\mathcal{H}_{\mathbf{x}_f} = [\mathbf{h}(\mathbf{x}_f, \mathbf{y}_{f,1}), \ldots, \mathbf{h}(\mathbf{x}_f, \mathbf{y}_{f,N})]', \ \forall \mathbf{y}_f \ \epsilon \ \mathrm{n-best}(\mathbf{x}_f) \tag{A.2}$$

Additionally, let $\mathcal{H}^*_{\mathbf{x}_f}$ be a matrix such that

$$\mathcal{H}^*_{\mathbf{x}_f} = [\mathbf{h}(\mathbf{x}_f, \mathbf{y}^*_f), \ldots, \mathbf{h}(\mathbf{x}_f, \mathbf{y}^*_f)] \tag{A.3}$$

where all rows are identical and equal to the feature vector of the best hypothesis $\mathbf{y}^*$ within the n-best list. Then, $R_{\mathbf{x}}$ is defined as:

$$R_{\mathbf{x}_f} = \mathcal{H}^*_{\mathbf{x}_f} - \mathcal{H}_{\mathbf{x}_f} \tag{A.4}$$

The key idea is to find out a vector $\check{\boldsymbol{\lambda}}$ such that differences in scores are reflected as differences in the hypotheses quality. That is:

$$R_{\mathbf{x}_f} \cdot \check{\boldsymbol{\lambda}} \propto \mathbf{l}_{\mathbf{x}_f} \tag{A.5}$$

where $\mathbf{l}_{\mathbf{x}_f}$ is a column vector of N rows such that:

$$\mathbf{l}_{\mathbf{x}_f} = [l(\mathbf{y}_{f,1}), ..., l(\mathbf{y}_{f,N})]', \forall \mathbf{y} \in \text{n} - \text{best}(\mathbf{x}_f) \tag{A.6}$$

The objective is to find $\check{\boldsymbol{\lambda}}^f$ such that:

$$\check{\boldsymbol{\lambda}}^f = \underset{\boldsymbol{\lambda}}{\text{argmin}} \, |\mathbf{R}_{\mathbf{x}_f} \cdot \boldsymbol{\lambda} - \mathbf{l}_{\mathbf{x}_f}| = \underset{\boldsymbol{\lambda}}{\text{argmin}} \, ||\mathbf{R}_{\mathbf{x}_f} \cdot \boldsymbol{\lambda} - \mathbf{l}_{\mathbf{x}_f}||^2 \tag{A.7}$$

where $\| \cdot \|^2$ is the Euclidean norm. Although both optimisations in Equation A.7 are equivalent (i.e., the $\check{\boldsymbol{\lambda}}^f$ that minimizes the first one also minimizes the second one), the second optimisation in Equation A.7 allows a direct implementation thanks to ridge regression. $\check{\boldsymbol{\lambda}}^f$ can be computed as the solution to the overdetermined system $R_{\mathbf{x}_f} \cdot \check{\boldsymbol{\lambda}}^f = \mathbf{l}_{\mathbf{x}_f}$, which is given by

$$\check{\boldsymbol{\lambda}}^f = (\mathbf{R}'_{\mathbf{x}_f} \cdot \mathbf{R}_{\mathbf{x}_f} + \beta \mathbf{I})^{-1} \cdot \mathbf{l}_{\mathbf{x}_f} \tag{A.8}$$

where a small β is used as a regularization term to stabilize $\mathbf{R}'_{\mathbf{x}_f} \cdot \mathbf{R}_{\mathbf{x}_f}$ and to ensure that it is invertible.

Algorithm 7 shows the pseudo-code for obtaining $\check{\boldsymbol{\lambda}}^f$. In this work, we apply the original DRR approach proposed by [190] to an off-line scenario, so that the method proposed is effectively able to compete with state-of-the-art $\boldsymbol{\lambda}$ estimation approaches. In this case, DRR obtains an estimation of $\boldsymbol{\lambda}$ by previously adjusting the $\boldsymbol{\lambda}$ vector to each of the sentences in a development corpus, i.e., the optimal $\boldsymbol{\lambda}$ is computed after performing a complete epoch on the development set.

A.2.2 BATCH DRR

The second DRR alternative for an off-line scenario is batch variation. Algorithm 8 shows the difference regarding the previous algorithm using, in this case, the batch version.

for *each of the sentences* $\mathbf{x}_f$ *in* $\mathcal{A}$ **do**
  $\mathcal{H}_{\mathbf{x}_f} \leftarrow [\mathbf{h}(\mathbf{x}_f, \mathbf{y}_{f,1}), \ldots, \mathbf{h}(\mathbf{x}_f, \mathbf{y}_{f,N})]'$;
  $\mathcal{H}^*_{\mathbf{x}_f} \leftarrow [\mathbf{h}(\mathbf{x}_f, \mathbf{y}^*_f), \ldots, \mathbf{h}(\mathbf{x}_f, \mathbf{y}^*_f)]'$;
  $R_{\mathbf{x}_f} \leftarrow \mathcal{H}_{\mathbf{x}^*_f} - \mathcal{H}_{\mathbf{x}_f}$;
  $\check{\boldsymbol{\lambda}}^f \leftarrow (\mathbf{R}'_{\mathbf{x}_f} \cdot \mathbf{R}_{\mathbf{x}_f} + \beta\mathbf{I})^{-1} \cdot \mathbf{l}_{\mathbf{x}_f}$;
  $\boldsymbol{\lambda}^f \leftarrow (1-\alpha)\boldsymbol{\lambda}^{f-1} + \alpha\check{\boldsymbol{\lambda}}^f$
end

Algorithm 7: Pseudo-code for computing the vector $\boldsymbol{\lambda}^f$ as described in Section A.2.1.1

Data: Development corpus $\mathcal{A}$
Result: $\boldsymbol{\lambda}$
Initialize: $\boldsymbol{\lambda}^0$;
forall *desired number of iterations* $\mathcal{I}$ **do**
  forall *number of batch* $b \in |\mathcal{B}|$ **do**
    optimization: compute vector $\check{\boldsymbol{\lambda}}^b_i$;
    estimation: $\boldsymbol{\lambda}^b_i = (1-\alpha)\boldsymbol{\lambda}^{b-1}_i + \alpha\check{\boldsymbol{\lambda}}^b_i$;
  end
end
selection: output vector $\boldsymbol{\lambda}^{\mathcal{B}}_I$

Algorithm 8: Pseudo-code for DRR estimating $\boldsymbol{\lambda}$ using a set of batches $\mathcal{B}$, as described in Section A.2.2

We have established a set of batches $\mathcal{B}$, with $A = \bigcup_{k=1}^{|\mathcal{B}|} b_k$, next term k has been omitted to simplify notation.

The main difference between sentence-to-sentence DRR and batch DRR relies in the **optimization**. In Algorithm 8, the **optimization** step estimates the vector $\check{\boldsymbol{\lambda}}$ described in Algorithm 7 for each of the development sentences $\mathbf{x}_c$, but the vector is estimated using all the information in subset b. In next section, we present the modification of the optimization step to account for this variation.

A.2.2.1 Batch-based optimisation in DRR

As exposed in the previous section, in DRR $\boldsymbol{\lambda}$ is calculated estimating the best value from all vectors obtained for each batch b. This algorithm is very similar to the one presented in Section A.2.1.1 yet, the algorithm uses all the information within batch b instead of only one sentence of the development corpus. For computing the new log-linear vector $\boldsymbol{\lambda}^b$, the previously learned $\boldsymbol{\lambda}^{b-1}$ needs to be combined with an appropriate update step $\check{\boldsymbol{\lambda}}^b$. The $\boldsymbol{\lambda}^b$ is calculated as a linear combination:

$$\boldsymbol{\lambda}^b = (1-\alpha)\boldsymbol{\lambda}^{b-1} + \alpha\check{\boldsymbol{\lambda}}^b \tag{A.9}$$

To obtain $\check{\boldsymbol{\lambda}}^b$, we changed the Equation A.8:

$$\check{\boldsymbol{\lambda}}^b = (\mathbf{R}_b' \cdot \mathbf{R}_b + \beta\mathbf{I})^{-1} \cdot \mathbf{l}_b \tag{A.10}$$

where $\mathbf{R}_b$ is defined as: $\mathbf{R}_b = \mathcal{H}_b^* - \mathcal{H}_b$. The matrix $\mathcal{H}_b$ $N \cdot |b| \times M$ contains the feature functions $\mathbf{h}$ of every hypothesis for all sentences $\mathbf{x} \in b$ and N is the size of n-best$(\cdot)$ for each sentence $\mathbf{x} \in b$:

$$\mathcal{H}_b = [\mathcal{H}_{\mathbf{x}_1}, \ldots, \mathcal{H}_{\mathbf{x}_{|b|}}]' \tag{A.11}$$

Additionally, $\mathcal{H}_M^*$ was redefined by:

$$\mathcal{H}_b^* = [\mathcal{H}_{\mathbf{x}_1}^*, \ldots, \mathcal{H}_{|b|}^*] \tag{A.12}$$

and $\mathbf{l}_b$ is a column vector of N rows such that:

$$\mathbf{l}_b = [l_{\mathbf{x}_1}, \ldots, l_{\mathbf{x}_{|b|}}]' \tag{A.13}$$

A.3 Experiments

In this section, we describe the experimental framework used to assess the performance of DRR variants described in Section A.2. We will first detail the experimental setup employed, and then, we will report the analysis of our method and it results. Finally, we will show a comparison between our DRR method and the two state-of-the-art optimisation methods: MERT and MIRA.

for *each of the batch b* **do**
  $\mathcal{H}_b \leftarrow [\mathcal{H}_{\mathbf{x}_1}, \ldots, \mathcal{H}_{\mathbf{x}_{|b|}}]'$;
  $\mathcal{H}_b^* \leftarrow [\mathcal{H}_{\mathbf{x}_1}, \ldots, \mathcal{H}^*_{\mathbf{x}_{|b|}}]'$;
  $R_b \leftarrow \mathcal{H}_b^* - \mathcal{H}_b$;
  $\check{\boldsymbol{\lambda}}^b \leftarrow (\mathbf{R}_b' \cdot \mathbf{R}_b + \beta \mathbf{I})^{-1} \cdot \mathbf{l}_b$;
  $\boldsymbol{\lambda}^f \leftarrow (1-\alpha)\boldsymbol{\lambda}^{b-1} + \alpha \check{\boldsymbol{\lambda}}^b$
end

Algorithm 9: Pseudo-code for computing the vector $\boldsymbol{\lambda}^b$ as described in Section A.2.2.1

A.3.1 CORPORA

We conducted experiments on two different language pairs: English-French (EN→ FR) and German-English (DE→ EN). Given that, the techniques described above are suited for adaptation purposes. We investigated the performance of these techniques in a cross-domain setting: we conducted experiments training the SMT system initially on an out-of-domain corpus and then analysing its performance on an in-domain corpus.

As out-of-domain corpus we used the Europarl corpus; details are given in Section 2.3. As in-domain corpus, we experimented with three different domains: Medical, News and Xerox. All details of these corpora were presented in Section 2.3, designed especially to describe the corpora used in this thesis.

A.3.2 EXPERIMENTAL SETUP

Experiments were performed using the phrase-based toolkit employed in this thesis, as describe in Section 2.4. We evaluated the translation system with three automatic metrics: BLEU, TER and METEOR, details in Section 2.2.

As described in Section A.2, the score function σ is correlated with some quality criterion. It is measured by some automatic metrics as BLEU or TER. We analysed the behaviour of DRR using both, BLEU and TER. We favour the use of BLEU because of its wider accepted in the SMT community. However, the original implementation of

BLEU is not always well defined at the sentence level. Given that, it implements a geometric average which is zero whenever there is no common 4-gram between the hypothesis and the reference, e.g. 3-word sentence. For this reason, we used smoothed BLEU, as defined by [192]. In the case of TER metric, the original work by [190] applied on-line DRR to optimise TER scores, which is why we decided to analyse it in this new scenario.

For each corpus, we trained baseline systems for comparison. This baseline was obtained by tuning the SMT system using an out-of-domain development corpus: the Europarl-Dev (corpus details in Table 2.2). We named this system *bsln*.

In these experiments, the confidence intervals are shown in different plots. Instead of using error bars, the translation quality plots will show vertical lines because otherwise, rendering will be unreadable.

A.3.3 DRR EXPERIMENTS

In this section, we present a study of our DRR variations. The following issues were studied:

1. Varying learning rate and n-best size.

2. Difference between sentence-by-sentence DRR and batch DRR.

In this study, we used only the development corpus of each domain since the purpose is to analyse the effect of adapting $\boldsymbol{\lambda}$. Accordingly, the results displayed are using the corresponding corpus for each domain (EMEA-DEV, NC-DEV and XRCE-DEV, details in Section 2.3). In addition, in all the experiments, the translation quality was measured using BLEU metric.

A.3.3.1 Varying learning rate and increasing the n-best size

As a first step, we analysed the effect of varying the learning rate α described in Equation A.1, together with different n-best sizes. Besides, we used the sentence-by-sentence DRR variation for these experiments.

Development corpus results are shown in Figure A.1 for each domain considered and using language pair EN→FR. We analysed a broad range of learning rates α, but we show only the most significant α values for clarity purposes. In addition, the translation quality obtained with the baseline system is also displayed. Several conclusions can be drawn:

- The results show that high values of α lead to a significant degradation in terms of translation quality. The reason behind can be explained by looking at Equation A.1. High values produce bigger changes of the λ^s respect to λ^{s-1}, and consequently an important change in the search space. On the contrary, smaller values of λ can obtain better translation quality.

- The effect of increasing the size of the n-best considered was also analysed. It can be seen that, the size of n-best and α are strongly related parameters. Higher α values need more n-best for obtaining better results. However, when the algorithm has the adequate learning rate, the n-best size does not have a large influence on the outcome in terms of translation quality.

A.3.3.2 Varying batch size

In this section, we compare the batch DRR (Section A.2.2) and sentence-by-sentence DRR (Section A.2.1). For this comparison, we conducted similar experiments to those in Section A.3.3.1 (varying α and n-best size), but we included the use of batches. The best results for each domain are presented in Table A.2. The results are shown considering BLEU as evaluation metric, and varying the number of batches $|\mathcal{B}|$. Table A.2 illustrates other important parameters to consider, $|b_k|$ which represents the number of sentences in each batch b_k and α, the learning rate used. Then, the first line in the Table represents the best result obtained with sentence-by-sentence DRR for each domain, as well as, the impact caused on the batch DRR variation compared to sentence-by-sentence DRR. We can observe similar results in terms of BLEU. Hence, we can conclude that both alternatives converge to a similar search space. The most remarkable difference is the learning

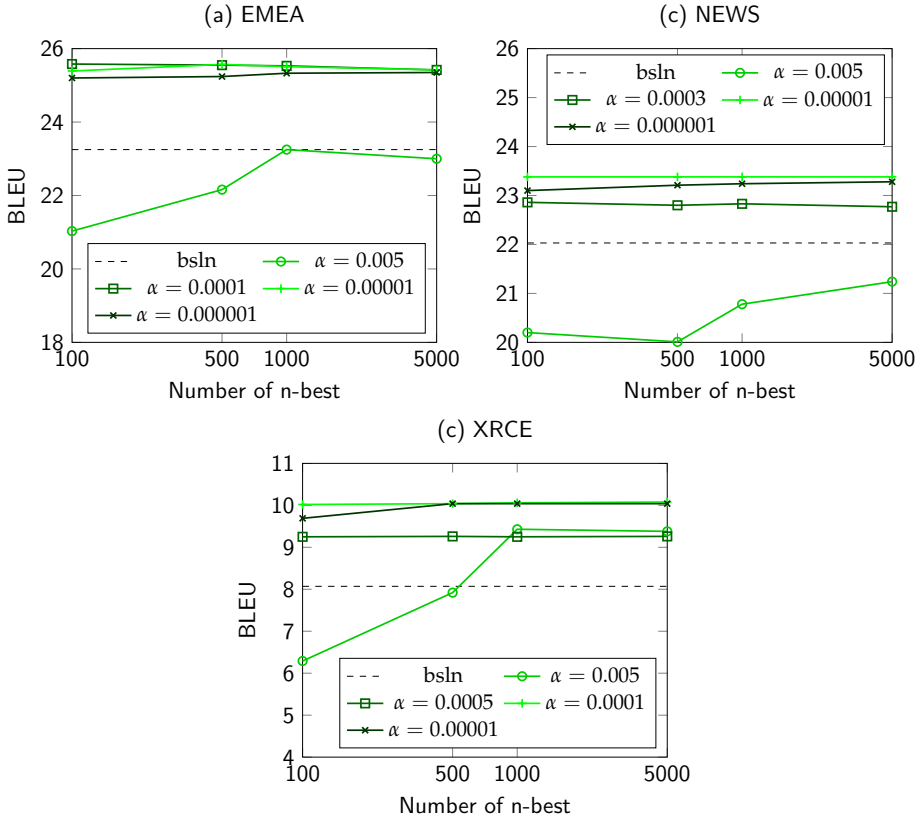


Figure A.1. Translation quality comparison considering different α values and n-best numbers.

rate value used to obtain the best results. In the case of sentence-by-sentence DRR, the best results are obtained with a very α small value. On the contrary, batch DRR obtained the same results with higher α values. We think this happens because of batch DRR includes more information in each update, and hence, the update steps are more stable.

A.3.4 Comparison between DRR, MERT and MIRA

Once the effect of the different parameters of DRR was analysed, we pursue to compare our method with standard methods such as MERT

Table A.2. Best results varying the batch size for each domain, evaluation made with BLEU.

	EN-FR				DE-EN			
Domain	$\mathcal{B}$	$\lvert b_k \rvert$	α	**BLEU**	$\mathcal{B}$	$\lvert b_k \rvert$	α	**BLEU**
EMEA	–	–	**0.00001**	**25.6**	–	–	**0.0005**	**19.6**
	3	240	0.01	25.5	3	240	0.01	19.4
	4	160	0.005	25.5	4	160	0.01	19.4
	7	80	0.005	25.5	7	80	0.001	19.4
NEWS	–	–	0.00001	23.3	–	–	0.00005	17.6
	4	400	0.005	23.4	4	400	0.01	17.5
	7	**200**	**0.001**	**23.5**	**7**	**200**	**0.001**	**17.6**
	28	50	0.001	23.5	28	50	0.0001	17.4
XRCE	–	–	0.0001	10.1	–	–	0.00005	10.3
	4	300	0.005	10.1	4	300	0.01	10.3
	7	150	0.0005	10.1	7	150	0.001	10.3
	20	50	0.0005	10.1	**20**	**50**	**0.001**	**10.4**

and MIRA. This comparative study was conducted taking into account the following issues:

1. Varying the size of the development corpus (Section A.3.4.1).

2. Increasing the number of n-best used within each method (Section A.3.4.2).

A.3.4.1 Size of the development set

As a first step in this comparative, we studied the effect of increasing the number of development samples available to the system. Figure A.2 and Figure A.3 show the effect of adding sentences to the development corpus and the confidence intervals derived.

These results show translation quality in terms of BLEU and TER (Figure A.2 for BLEU, Figure A.3 for TER), for each domain and development corpus considered (details in Section A.3.1). For clarity, we only show results for the best meta-parameter configuration of DRR obtained in Section A.3.3, and the standard parameter configuration of MERT and MIRA present in the Moses toolkit.

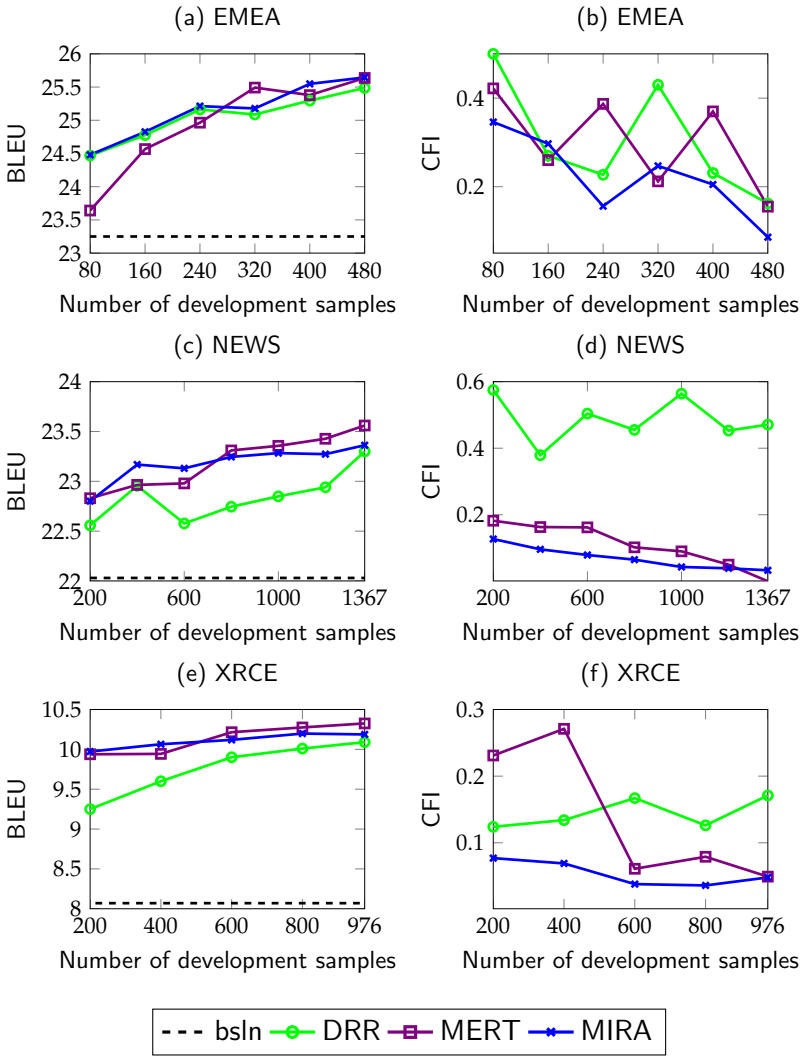


Figure A.2. Performance comparison across the different corpora analysed and different λ estimation methods. The three plots on the left display BLEU, while the three plots on the right display the size of the confidence intervals (CFI).

Results of such comparison can be seen in Figure A.2, in terms of BLEU. Some things should be noted:

- Results obtained for all methods are better than the baseline system (bsln) at the beginning, as could be expected. This could be demonstrating the effectiveness of these methods in an SMT adaptation task.

- All the methods have the same behaviour when the development size increases, leading to improvements in translation quality as measured by BLEU, (around $1-2$ points better).

- Results obtained with our DRR method are similar than the ones obtained with MIRA and MERT.

- As expected, smaller amounts of development corpus lead to larger confidence intervals.

- The NEWS corpus appears to be a specially difficult corpus for the DRR method. Also, confidence intervals are especially high when compared to the other methods.

- Lastly, when looking at the translation quality of the XRCE corpus, it stands out that curves behave especially poorly in terms of BLUE (around 10 points). The development set of the XRCE corpus seems to be quite different from the training data (Europarl training), which implies that the system is not able to obtain a good n-best list.

Since TER is another evaluation metric commonly used in the SMT community and was the metric used initially for on-line DRR, we also included it in our analysis. In general terms, the overall analysis is very similar concerning the one using BLEU. Figure A.3 shows the main results obtained for the three domains and EN-FR language pair.

- Using TER for estimating λ leads to a similar results as compared to BLEU.

- Increasing the number of adaptation samples leads to better results for all the methods considered, without being statistically significant in terms of BLEU.

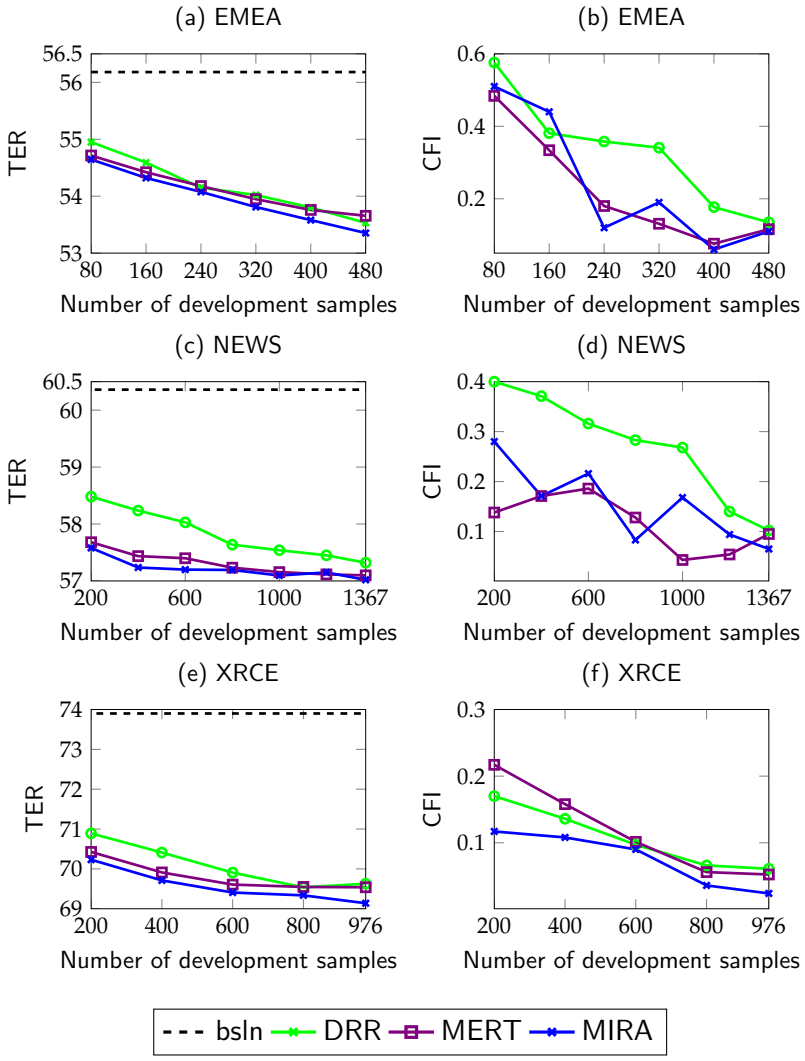


Figure A.3. Performance comparison across the different corpora analysed and with the different λ estimation methods. The tree plots on the left display TER, while the three plots on the right display the size of the confidence intervals (CFI).

- Similarly to previous experiments, DRR obtains weak results in the NEWS corpus. Nevertheless, the difference is not significant.

A.3.4.2 Variating n-best size

As explained before, all methods described before leverage a n-best list for optimizing the λ of the log-lineal model. In case of MERT, the n-best list is used to obtain the best hypotheses. In the case of DRR, the n-best list has greater importance to the algorithm; since it uses all the information contained in the list. For this reason, we analyse the impact of different n-best list sizes although we do not report all of them to avoid making this table too many large. We only report the best result obtained regarding the n-best size.

Table A.3 and Table A.4 show the best results in terms of BLEU, TER and METEOR. Some things should be noted:

- In terms of the metric used in Table A.3 and Table A.4 (BLEU or TER respectively) we can see that all the methods yield very similar results without statistical significance. These results are consistent across all different domains and language pairs.
- If we focus our analysis on n-best size, the DRR method needs more n-best to obtain better results, as compared to the other two methods. This should be expected, since MERT and MIRA conduct several translation steps, whereas DRR only conducts one single step. Hence, for a fixed n-best size, MERT and MIRA have access to more information than DRR, since they re-compute the search space after each translation step. DRR was actually designed to have access to a large n-best list.
- We analyse the difference between metrics, BLEU and TER. We can note a difference in the results for the same metric in the two Tables. These differences have been expected due to the method is focused on optimising a specific metric. To provide a concrete example, in the Tables A.3 and A.4, BLEU metric has a difference of around $2-3$ points for DRR method.

A.3.4.3 Results analysis for test corpus and computational time

Given that tuning is critical for adapting a PBSMT system to a specific domain or corpus, in this section we present the best results ob-

Table A.3. Impact caused by changing the n-best list size for each method. Translation quality measured in terms of BLEU. 0.1^* represents confidence intervals that are less than 0.1

Language	Domain	Method	|n − best|	**BLEU**	**TER**	**METEOR**
EN-FR	EMEA	MERT	**200**	$\mathbf{25.7 \pm 0.1}$	$\mathbf{56.1 \pm 0.5}$	$\mathbf{48.6 \pm 0.2}$
		MIRA	200	25.6 ± 0.1	55.8 ± 0.2	48.6 ± 0.1
		DRR	100	25.5 ± 0.2	55.8 ± 0.3	48.5 ± 0.1
	NEWS	MERT	**100**	$\mathbf{23.5 \pm 0.1}$	$\mathbf{59.3 \pm 0.1^*}$	$\mathbf{47.5 \pm 0.1}$
		MIRA	200	$23.4 \pm 0.1^*$	59.2 ± 0.8	$47.5 \pm 0.1^*$
		DRR	500	23.2 ± 0.5	58.5 ± 0.7	46.1 ± 0.7
	XRCE	MERT	**200**	$\mathbf{10.3 \pm 0.1^*}$	$\mathbf{74.2 \pm 0.6}$	$\mathbf{34.2 \pm 0.3}$
		MIRA	200	$10.2 \pm 0.1^*$	74.0 ± 0.2	34.0 ± 0.1
		DRR	500	10.1 ± 0.1	73.7 ± 0.2	33.7 ± 0.1
DE-EN	EMEA	MERT	500	19.8 ± 0.1	60.6 ± 0.4	26.6 ± 0.1
		MIRA	**100**	$\mathbf{19.7 \pm 0.1}$	$\mathbf{60.6 \pm 0.2}$	$\mathbf{26.6 \pm 0.3}$
		DRR	200	19.4 ± 0.3	60.0 ± 0.4	26.6 ± 0.1
	NEWS	MERT	**200**	$\mathbf{18.1 \pm 0.1^*}$	$\mathbf{65.8 \pm 0.1}$	$\mathbf{27.4 \pm 0.1^*}$
		MIRA	200	$17.9 \pm 0.1^*$	65.3 ± 0.1	$27.3 \pm 0.1^*$
		DRR	500	17.5 ± 0.2	66.7 ± 1.1	27.3 ± 0.1
	XRCE	MERT	**500**	$\mathbf{11.3 \pm 0.1^*}$	$\mathbf{71.9 \pm 1.4}$	$\mathbf{17.6 \pm 0.2}$
		MIRA	200	10.8 ± 0.2	72.1 ± 1.3	$17.5 \pm 0.1^*$
		DRR	500	10.4 ± 0.2	69.6 ± 0.3	17.4 ± 0.1

tained for each method when translating different domains (Medical-test, NC-test and XRCE-test). These corpora have only been used for optimizing the log-linear weights so, the only information that the system has was obtained during the tuning process using the development corpus that belongs to the same domain as the test corpus.

The vector weights $\boldsymbol{\lambda}$ used in each method were obtained using the best configuration analysed in previous sections. In addition, we reported in Sections A.3.4.1 and A.3.4.2 that the methods analysed have a similar behaviour in terms of BLEU or TER. For this reason, the results using TER were removed from the final comparison in order to avoid clogging the chapter with too many similar tables.

In Table A.5, we show the main results obtained for each method (MERT, MIRA and DRR) in terms of the three metrics studied (BLEU, TER and METEOR). As shown, our method is able to yield slightly

Table A.4. Impact caused by changing the n-best list size for each method. Translation quality measured in terms of TER. 0.1^* represents confidence intervals that are less than 0.1

Language	Domain	Method	$\lvert n-best \rvert$	**TER**	**BLEU**	**METEOR**
EN-FR	EMEA	MERT	200	53.7 ± 0.1	24.4 ± 0.2	47.2 ± 0.3
		MIRA	**200**	$\mathbf{53.4 \pm 0.1}$	$\mathbf{24.7 \pm 0.2}$	$\mathbf{47.2 \pm 0.1}$
		DRR	500	53.6 ± 0.2	24.7 ± 0.3	47.1 ± 0.1
	NEWS	MERT	200	57.1 ± 0.1	21.5 ± 0.2	44.8 ± 0.2
		MIRA	**200**	$\mathbf{57.1 \pm 0.1}^*$	$\mathbf{21.3 \pm 0.4}$	$\mathbf{44.8 \pm 0.1}$
		DRR	1000	57.5 ± 0.4	20.7 ± 0.7	44.0 ± 0.6
	XRCE	MERT	1000	69.4 ± 0.1	8.7 ± 0.1	30.8 ± 0.2
		MIRA	**500**	$\mathbf{69.1 \pm 0.1}$	$\mathbf{8.9 \pm 0.1}$	$\mathbf{31.0 \pm 0.2}$
		DRR	200	69.5 ± 0.1	8.7 ± 0.2	30.7 ± 0.3
DE-EN	EMEA	MERT	200	57.5 ± 0.1	17.4 ± 0.3	25.4 ± 0.1
		MIRA	**200**	$\mathbf{57.3 \pm 0.1}$	$\mathbf{17.6 \pm 0.2}$	$\mathbf{25.5 \pm 0.2}$
		DRR	500	57.6 ± 0.3	17.9 ± 0.4	25.4 ± 0.1
	NEWS	MERT	**100**	$\mathbf{62.3 \pm 0.1}^*$	$\mathbf{16.0 \pm 0.2}$	$\mathbf{26.2 \pm 0.1}$
		MIRA	200	$62.6 \pm 0.1^*$	16.3 ± 0.1	26.1 ± 0.1
		DRR	500	63.0 ± 0.2	15.8 ± 0.9	26.1 ± 0.3
	XRCE	MERT	**200**	$\mathbf{66.8 \pm 0.1}$	$\mathbf{9.1 \pm 0.4}$	$\mathbf{16.4 \pm 0.1}$
		MIRA	200	67.2 ± 0.1	8.8 ± 0.4	16.3 ± 0.1
		DRR	200	67.1 ± 0.1	9.0 ± 0.4	16.2 ± 0.1

better results in most domains (EMEA DE-EN, XRCE EN-FR and DE-EN), although differences are not statistical significant. In the other cases, our DRR method leads to competitive results with respect to the state-of-the-art methods. We believe that this is important, since it proves the competitiveness of our proposal in this task, regarding other techniques that have been largely studied by the SMT community and are considered the state-of-the-art. In terms of computational time, Table A.6 reports the time consumed by each of the approaches reported in Table A.5. All experiments were performed under the same conditions and computers. Computational time was measured in single-threaded runs of the algorithms running on a 64 bit machine with Intel Xeon CPUs at 2.50 GHz with 6 MB cache. As shown, the DRR method is faster than the other two methods, while still better or similar results are obtained.

Table A.5. Best translation results for the tree domains using different optimization algorithm using BLEU metric. 0.1^* represent to confidence intervals whose value is less than 0.1

Language	Domain	Method	BLUE	TER	METEOR
EN-FR	EMEA	MERT	22.3 ± 0.2	58.8 ± 0.1	45.0 ± 0.2
		MIRA	$\mathbf{22.4 \pm 0.1}$	$\mathbf{58.5 \pm 0.2}$	$\mathbf{45.1 \pm 0.1}$
		DRR	22.3 ± 0.2	58.5 ± 0.2	45.0 ± 0.2
	NEWS	MERT	27.0 ± 0.1	53.6 ± 0.1	50.8 ± 0.2
		MIRA	$\mathbf{27.4 \pm 0.1}$	$\mathbf{53.0 \pm 0.1}$	$\mathbf{51.1 \pm 0.1}$
		DRR	26.8 ± 0.3	54.3 ± 0.9	51.0 ± 0.2
	XRCE	MERT	9.9 ± 0.3	73.6 ± 0.8	36.8 ± 0.3
		MIRA	10.2 ± 0.1	73.3 ± 0.2	36.1 ± 0.1
		DRR	$\mathbf{10.2 \pm 0.1}$	$\mathbf{73.0 \pm 0.2}$	$\mathbf{36.7 \pm 0.1}$
DE-EN	EMEA	MERT	19.1 ± 0.2	61.7 ± 0.5	27.1 ± 0.1
		MIRA	19.1 ± 0.1	61.4 ± 0.2	27.2 ± 0.1
		DRR	$\mathbf{19.2 \pm 0.2}$	$\mathbf{60.4 \pm 0.5}$	$\mathbf{27.2 \pm 0.1}$
	NEWS	MERT	21.6 ± 0.1	58.5 ± 0.1	$30.4 \pm 0.1^*$
		MIRA	$\mathbf{21.7 \pm 0.1^*}$	$\mathbf{57.9 \pm 0.1^*}$	$\mathbf{30.5 \pm 0.1^*}$
		DRR	21.1 ± 0.3	58.6 ± 1.0	30.2 ± 0.1
	XRCE	MERT	9.4 ± 0.3	73.9 ± 0.9	18.1 ± 0.2
		MIRA	9.8 ± 0.1	72.5 ± 1.0	18.1 ± 0.1
		DRR	$\mathbf{9.6 \pm 0.1}$	$\mathbf{70.1 \pm 0.4}$	$\mathbf{18.0 \pm 0.1}$

Table A.6. Time consumed by the different approaches compared. Results are given in minutes.

		EN-FR	DE-EN
Domain	Method	Computational Time	Computational Time
EMEA	MERT	252	240
	MIRA	300	350
	DRR	**190**	**200**
NC	MERT	480	900
	MIRA	660	840
	DRR	**432**	**720**
XRCE	MERT	400	390
	MIRA	420	390
	DRR	**360**	**350**

A.4 Summary

In this chapter, the DRR method has been thoroughly analysed regarding its application to log-linear vector weight adaptation in SMT. On the one hand, the theoretical framework for adapting the log-linear weights present in phrases-based SMT systems has been developed. On the other hand, experimental results analysing the effectiveness of such adaptation method have been reported.

Results show that DRR has an interesting potential in the adaptation task. Consistent improvements in translation quality are obtained over the baseline system measured by BLEU and TER metrics. We have demonstrated, via empirical experiments, that our DRR method obtains comparable or better results than MERT and MIRA, including a reduction of computational time across different domains and language pairs. We consider this fact important, since it means that DRR is able to lead to competitive results while using less computational resources.

List of Symbols and Abbreviations

Abbreviation	Description	Definition
MT	Machine Translation	page xiii
SMT	Statistical Machine Translation	page xiii
NMT	Neural Machine Translation	page xiv
NLP	Natural Language Processing	page 2
RBMT	Rule-Based Machine Translation	page 3
$\mathbf{x}$	source sentence	page 4
$\mathbf{y}$	target sentence	page 4
$\mathcal{X}$	source vocabularies	page 4
$\mathcal{Y}$	target vocabularies	page 4
$J = \lvert\mathbf{x}\rvert$	Lengths of the source sentence for $\mathbf{x}$	page 4
$I = \lvert\mathbf{y}\rvert$	Lengths of the target sentence for $\mathbf{y}$	page 4
$\hat{\mathbf{y}}$	estimated sentence	page 4
$\mathbf{a}$	IBM models alignment	page 5
$p(\mathbf{x}, a\lvert\mathbf{y})$	probability translation given a	page 5
M	number of models in the log-linear model	page 5
$h_m(\mathbf{x}, \mathbf{y})$	score function	page 5
λ_m	weight for $h_m(\mathbf{x}, \mathbf{y})$	page 5
$\boldsymbol{\lambda}$	$\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_M]$	page 6
PB	phrase-based	page 6
K	number of phrases	page 6
MERT	minimum error rate training	page 9
MIRA	margin infused relaxed algorithm	page 9
$\mathbf{k}_\mathbf{x}$	represent which words have been translated	page 11
LM	language model	page 12
n	order of the n-gram	page 12
RNN	Recurrent Neural Network	page 21
BLEU	measure of precision	page 28
METEOR	METEOR evaluation metric	page 28
TER	measure of error	page 28

Abbreviation	Description	Definition
$\|S\|$	number of sentences	page 30
$\|W\|$	number of words (tokens)	page 30
V	vocabulary	page 30
$\|V\|$	vocabulary size	page 30
WMT	Workshop on Statistical Machine Translation	page 31
IT	Information Technology	page 32
Moses	decoder	page 37
GIZA++	phrase table toolkit	page 37
SRILM	language model toolkit	page 37
DS	Data Selection	page 43
TF-IDF	Term Frequency, Inverse Document Frequency	page 44
$H_C(\mathbf{x})$	cross-entropy given $\mathbf{x}$	page 46
D	source in-domain corpus	page 46
G	source out-of-domain corpus	page 46
$H_D(\mathbf{x})$	cross-entropy given $\mathbf{x}$ from D	page 46
$H_G(\mathbf{x})$	cross-entropy given $\mathbf{x}$ from G	page 46
$c(\mathbf{x})$	cross-entropy difference score of $\mathbf{x}$	page 46
E	target in-domain corpus	page 46
L	target out-of-domain corpus	page 46
$c(\mathbf{x}, \mathbf{y})$	bilingual cross-entropy difference score	page 46
T	source test corpus	page 47
X	set of n-grams that appear in a sentences	page 47
m	n-gram in X	page 47
$R(m)$	counts of m in a given $\mathbf{x}$ of the G	page 47
$C(m)$	counts of m in a given $\mathbf{x}$ of the D	page 47
t	frequency n-grams term	page 47
$i(\mathbf{x})$	infrequency score for sentence $\mathbf{x}$	page 47
$\mathbf{x}^*$	$\mathbf{x}$ with the highest score $i(\mathbf{x})$	page 47
$i(\mathbf{x}^*)$	infrequency score for sentence $\mathbf{x}^*$	page 47
CVR	continuous vector-space representation	page 14
size	vector dimensions	page 16
CBOW	Continuous Bag of Words Model	page 16
Skip-Gram	Continuous Skip-Gram Model	page 16
word2vec	word embeddings toolkit	page 37
$f(w)$	word embeddings	page 18
$F_{\mathbf{x}}$	sentence embeddings for $\mathbf{x}$	page 18
$N_{\mathbf{x}}$	count of w in sentence $\mathbf{x}$	page 19
`Mean-vec`	sentence embedding method	page 18
`Document-vec`	sentence embedding method	page 18
`Selected-corpus`	selected-corpus created with some DS method	page 54
S	similarity corpus	page 53

Abbreviation	Description	Definition
$sim_i(\cdot,\cdot)$	similarity function	page 54
τ	similarity threshold	page 54
$cos(\cdot,\cdot)$	cosine similarity	page 54
CNNs	Convolutional neural networks	page 57
LSTM	Bidirectional LSTM networks	page 58
ρ_G	out-of-domain phrase tables	page 89
ρ_D	in-domain phrase tables	page 89
ρ_F	final phrase table obtained by fill-up method	page 89
DDS	Development Data Selection	page 105
LD	Levenshtein Distance	page 106
LD-DDS	Levenshtein Distance for DDS method	page 106
P	pool of development sentences available	page 106
`Dev-Corpus`	selected development corpus	page 106
$F_{\mathbf{T}}$	mean for test corpus T	page 108
TF-DDS	DDS technique that derives from using TF-IDF	page 110
CVR-DDS	DDS technique that derives from using CVR	page 110
BPE	Byte pair encoding	page 125
A	Bilingual development corpus	page 143
σ	Scoring function for DRR method	page 143
$\mathcal{I}$	Maximum number of iterations	page 143
$\check{\lambda}^c$	Update term of DRR method	page 145
$\mathcal{B}$	Number of batch for Batch DRR method	page 146

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] W. J. Hutchins, *Machine Translation: Past, Present, Future*. New York, NY, USA: John Wiley & Sons, Inc., 1986.

[2] M. L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J. A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. M. Tyers, "Apertium: a free/open-source platform for rule-based machine translation," *Machine translation*, vol. 25, no. 2, pp. 127–144, 2011.

[3] F. M. Tyers, H. A. i Font, G. Fronteddu, and A. Martín-Mor, "Rule-based machine translation for the italian–sardinian language pair," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, no. 1, pp. 221–232, 2017.

[4] E. Bick, "Dan2eng: wide-coverage danish-english machine translation," *Proceedings of Machine Translation Summit*, pp. 37–43, 2007.

[5] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2010.

[6] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, pp. 263–311, 1993.

[7] T. Bayes, "An essay towards solving a problem in the doctrine of chances," *Philosophical Transaction of the Royal Society of London*, pp. 370–418, 1763.

[8] F. J. Och and H. Ney, "The alignment template approach to statistical machine translation," *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.

[9] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, pp. 39–71, 1996.

[10] K. A. Papineni, S. Roukos, and R. T. Ward, "Maximum likelihood and discriminative training of direct translation models," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 189–192, 1998.

[11] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pp. 295–302, 2002.

[12] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 48–54, 2003.

[13] J. Tomás and F. Casacuberta, "Monotone statistical translation using word groups," in *Proceedings of the Machine Translation Summit*, pp. 357–361, 2001.

[14] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *Proceedings of the Annual Conference on Artificial Intelligence*, pp. 18–32, 2002.

[15] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[16] F. J. Och, *Statistical machine translation: from single-word models to alignment templates*. PhD thesis, RWTH Aachen, 2002.

[17] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pp. 160–167, 2003.

[18] G. Gascó, V. Alabau, J. Andrés-Ferrer, J. González-Rubio, M.-A. Rocha, G. Sanchis-Trilles, F. Casacuberta, J. González, and J.-A. Sánchez, "ITI-UPV system description for IWSLT 2010.," in *Proceedings of International Workshop on Spoken Language Translation*, pp. 85–92, 2010.

[19] G. Sanchis-Trilles and F. Casacuberta, "Log-linear weight optimisation via bayesian adaptation in statistical machine translation," in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pp. 1077–1085, 2010.

[20] J. H. Clark, C. Dyer, A. Lavie, and N. A. Smith, "Better hypothesis testing for statistical machine translation: Controlling for optimizer instability," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 176–181, 2011.

[21] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[22] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "Online large-margin training for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 303–310, 2007.

[23] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pp. 427–436, 2012.

[24] A. Sokolov and F. Yvon, "Minimum error rate training semiring," in *Proceedings of the Annual Conference of the European Association for Machine Translation*, pp. 241–248, 2011.

[25] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1352–1362, 2011.

[26] B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, "Max-margin parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1–8, 2004.

[27] K. Knight, "Decoding complexity in word-replacement translation models," *Computational Linguistics*, vol. 25, pp. 607–615, 1999.

[28] D. Ortiz, *Advances in fully-automatic and interactive phrase-based statistical machine translation*. PhD thesis, Universitat Politècnica de València, 2011.

[29] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada, "Fast decoding and optimal decoding for machine translation," in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pp. 228–235, 2001.

[30] Y.-Y. Wang, *Grammar inference and statistical machine translation*. PhD thesis, Carnegie Mellon University Pittsburgh, PA, 1998.

[31] C. Tillmann, *Word re-ordering and dynamic programming based search algorithm for statistical machine translation*. PhD thesis, RWTH Aachen, 2001.

[32] C. Tillmann and H. Ney, "Word reordering and a dynamic programming beam search algorithm for statistical machine translation," *Computational Linguistics*, vol. 29, pp. 97–133, 2003.

[33] F. Jelinek, *Statistical methods for speech recognition*. MIT press, 1997.

[34] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 2, pp. 179–190, 1983.

[35] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pp. 310–318, 1996.

[36] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 181–184, 1995.

[37] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?," *Proceedings of the IEEE*, vol. 88, pp. 1270–1278, 2000.

[38] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker, "Perplexity—a measure of the difficulty of speech recognition tasks," *The Journal of the Acoustical Society of America*, vol. 62, pp. S63–S63, 1977.

[39] G. E. Hinton, "Learning distributed representations of concepts," in *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 12–24, 1986.

[40] J. L. McClelland, D. E. Rumelhart, *et al.*, *Parallel distributed processing*, vol. 2. Cambridge University Press, 1987.

[41] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[42] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[43] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proceedings of the International Conference on Machine Learning*, pp. 513–520, 2011.

[44] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 151–161, 2011.

[45] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches." *arXiv:1409.1259*, 2014.

[46] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models." *arXiv:1612.03651*, 2016.

[47] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[48] T. Mikolov, M. Karafiát, L. Burget, J. Eernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the INTERSPEECH*, pp. 1045–1048, 2010.

[49] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space." *arXiv:1301.3781*, 2013.

[50] R. Paulus, R. Socher, and C. D. Manning, "Global belief recursive neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2888–2896, 2014.

[51] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation.," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

[52] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[53] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the International Conference on Machine Learning*, pp. 160–167, 2008.

[54] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models." *arXiv:1206.6426*, 2012.

[55] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[56] J. Guo, W. Che, H. Wang, and T. Liu, "Learning sense-specific word embeddings by exploiting bilingual resources.," in *Proceedings of the International Conference on Computational Linguistics*, pp. 497–507, 2014.

[57] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 873–882, 2012.

[58] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space." *arXiv:1504.06654*, 2015.

[59] X. Yang and K. Mao, "Learning multi-prototype word embedding from single-prototype word embedding with integrated knowledge," *Expert Systems with Applications*, vol. 56, pp. 291–299, 2016.

[60] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method." *arXiv:1402.3722*, 2014.

[61] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," *Journal of Machine Learning Research*, vol. 13, pp. 307–361, 2012.

[62] J. Mitchell and M. Lapata, "Composition in distributional models of semantics," *Cognitive Science*, vol. 34, no. 8, pp. 1388–1429, 2010.

[63] Y. Kim, "Convolutional neural networks for sentence classification." *arXiv:1408.5882*, 2014.

[64] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." *arXiv:1405.4053*, 2014.

[65] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014.

[66] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi, "Extractive summarization using continuous vector space models," in *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pp. 31–39, 2014.

[67] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

[68] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, pp. 1–10, 2015.

[69] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg, "Fine-grained analysis of sentence embeddings using auxiliary prediction tasks." *arXiv:1608.04207*, 2016.

[70] H. Schwenk, A. Rousseau, and M. Attik, "Large, pruned or continuous space language models on a gpu for statistical machine translation," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 11–19, 2012.

[71] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models.," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 413–425, 2013.

[72] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pp. 1–10, 2015.

[73] Y. Tang, F. Meng, Z. Lu, H. Li, and P. L. Yu, "Neural machine translation with external phrase memory." *arXiv:1606.01792*, 2016.

[74] S. Shen, Y. Cheng, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Minimum risk training for neural machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1683–1692, 2016.

[75] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, "Deep recurrent models with fast-forward connections for neural machine translation." *arXiv:1606.04199*, 2016.

[76] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, "Neural versus phrase-based machine translation quality: a case study," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 257–267, 2016.

[77] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning." *arXiv:1705.03122*, 2017.

[78] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv:1412.3555*, 2014.

[79] P. Koehn, "Neural machine translation." *arXiv:1709.07809*, 2017.

[80] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proccedings of the Annual Meeting on Association for Computational Linguistics*, pp. 311–318, 2002.

[81] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proccedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and Summarization*, pp. 65–72, 2005.

[82] M. D. A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proccedings of the Annual Meeting on Association for Computational Linguistics*, pp. 376–387, 2014.

[83] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proccedings of the Annual Meeting on Association for Machine Translation in the Americas*, pp. 223–231, 2006.

[84] P. Koehn, "Statistical significance tests for machine translation evaluation.," in *Proccedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 388–395, 2004.

[85] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of the Machine Translation Summit*, pp. 79–86, 2005.

[86] U. Germann, "Aligned hansards of the 36th parliament of canada, release 2001-1a," 2001.

[87] M. Carpuat, H. D. III, A. Fraser, C. Quirk, F. Braune, A. Clifton, A. Irvine, J. Jagarlamudi, J. Morgan, M. Razmara, A. Tamchyna, K. Henry, and R. Rudinger, "Domain adaptation in machine translation: Final report," in *Johns Hopkins Summer Workshop Final Report*, Johns Hopkins University, 2012.

[88] J. Tiedemann, "News from opus - a collection of multilingual parallel corpora with tools and interfaces," in *Proceedings of the Recent Advances in Natural Language Processing*, pp. 237–248, 2009.

[89] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, and L. Specia, eds., *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2014.

[90] J. Tiedemann, "Parallel data, tools and interfaces in opus," in *Proceedings of the Language Resources and Evaluation Conference*, pp. 2214–2218, 2012.

[91] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, *et al.*, "Findings of the 2016 conference on machine translation," in *Proceedings of the First Conference on Machine Translation*, pp. 131–198, 2016.

[92] S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomás, E. Vidal, *et al.*, "Statistical approaches to computer-assisted translation," *Computational Linguistics*, vol. 35, no. 1, pp. 3–28, 2009.

[93] J. R. Smith, H. Saint-Amand, M. Plamada, P. Koehn, C. Callison-Burch, and A. Lopez, "Dirt cheap web-scale parallel text from the common crawl.," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1374–1383, 2013.

[94] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen, "The united nations parallel corpus v1.0," in *Proceedings of the Conference on Language Resources and Evaluation*, pp. 3530–3534, 2016.

[95] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling." *arXiv:1312.3005*, 2013.

[96] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proccedings of the Annual Meeting on Association for Computational Linguistics*, pp. 177–180, 2007.

[97] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proccedings of the International Conference on Spoken Language Processing*, pp. 901–904, 2002.

[98] A. Peris, "NMT-Keras." https://github.com/lvapeab/nmt-keras, 2017. GitHub repository.

[99] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[100] I. Redko and Y. Bennani, "Non-negative embedding for fully unsupervised domain adaptation," *Pattern Recognition Letters*, vol. 77, pp. 35–41, 2016.

[101] A. Axelrod, *Data Selection for Statistical Machine Translation*. PhD thesis, University of Washington, 2014.

[102] H. Cuong and K. Sima'an, "A survey of domain adaptation for statistical machine translation," *Machine Translation*, vol. 31, no. 4, pp. 187–224, 2017.

[103] G. Foster, B. Chen, and R. Kuhn, "Simulating discriminative training for linear mixture adaptation in statistical machine translation," *Proceedings of the Machine Translation Summit*, pp. 183–190, 2013.

[104] R. Sennrich, "Perplexity minimization for translation model domain adaptation in statistical machine translation," in *Proceedings of the European Chapter of the Association for Computational Linguistics*, pp. 539–549, 2012.

[105] G. Foster and R. Kuhn, "Mixture-model adaptation for smt," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 128–135, 2007.

[106] A. Finch and E. Sumita, "Dynamic model interpolation for statistical machine translation," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 208–215, 2008.

[107] J. Civera and A. Juan, "Domain adaptation in statistical machine translation with mixture modelling," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 177–180, 2007.

[108] N. Bertoldi and M. Federico, "Domain adaptation for statistical machine translation with monolingual resources," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 182–189, 2009.

[109] S. Abdul-Rauf, H. Schwenk, P. Lambert, and M. Nawaz, "Empirical use of information retrieval to build synthetic data for smt domain adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 745–754, 2016.

[110] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 86–96, 2016.

[111] P. Lambert, H. Schwenk, C. Servan, and S. Abdul-Rauf, "Investigations on translation model adaptation using monolingual data," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 284–293, 2011.

[112] H. Schwenk and J. Senellart, "Translation model adaptation for an arabic/french news translation system by lightly-supervised training," in *Proceedings of the Machine Translation Summit*, pp. 31–41, 2009.

[113] O. Bojar and A. Tamchyna, "Improving translation model by monolingual data," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 330–336, 2011.

[114] B. Marie and A. Fujita, "Phrase table induction using in-domain monolingual data for domain adaptation in statistical machine translation," *Transactions of the Association of Computational Linguistics*, vol. 5, no. 1, pp. 487–500, 2017.

[115] R. Sellami, F. Sadat, and L. H. Beluith, "Building and exploiting domain-specific comparable corpora for statistical machine translation," in *Proceedings of the Intelligent Natural Language: Trends and Applications*, pp. 659–676, 2018.

[116] B. Zhao, M. Eck, and S. Vogel, "Language model adaptation for statistical machine translation with structured query models," in *Proceedings of the International Conference on Computational Linguistics*, pp. 411–420, 2004.

[117] M. Eck, S. Vogel, and A. Waibel, "Language model adaptation for statistical machine translation based on information retrieval," in *Proceedings of the Conference on Language Resources and Evaluation*, pp. 327–330, 2004.

[118] P. Koehn and J. Schroeder, "Experiments in domain adaptation for statistical machine translation," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 224–227, 2007.

[119] A. Ceauşu, J. Tinsley, J. Zhang, and A. Way, "Experiments on domain adaptation for patent machine translation in the pluto project," in *Proceedings of the Annual Meeting of the European Association for Machine Translation*, pp. 21–29, 2011.

[120] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of domain adaptation methods for neural machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 385–391, 2017.

[121] P. Banerjee, S. K. Naskar, J. Roturier, A. Way, and J. van Genabith, "Translation quality-based supplementary data selection by incremental update of translation models.," in *Proceedings of the Conference on Computational Linguistics*, pp. 149–166, 2012.

[122] S. Eetemadi, W. Lewis, K. Toutanova, and H. Radha, "Survey of data-selection methods in statistical machine translation," *Machine Translation*, vol. 29, no. 3-4, pp. 189–223, 2015.

[123] A. S. Hildebrand, M. Eck, S. Vogel, and A. Waibel, "Adaptation of the translation model for statistical machine translation based on information retrieval," in *Proceedings of the Annual Conference of the European Association for Machine Translation*, pp. 133–142, 2005.

[124] Y. Lü, J. Huang, and Q. Liu, "Improving statistical machine translation performance by training data selection and optimization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 343–350, 2007.

[125] J. Gao, J. Goodman, M. Li, and K.-F. Lee, "Toward a unified approach to statistical language modeling for chinese," *ACM Transactions on Asian Language Information Processing*, vol. 1, no. 1, pp. 3–33, 2002.

[126] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pp. 220–224, 2010.

[127] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 355–362, 2011.

[128] A. Rousseau, "Xenc: An open-source tool for data selection in natural language processing," *The Prague Bulletin of Mathematical Linguistics*, vol. 100, pp. 73–82, 2013.

[129] A. Axelrod, X. He, P. Resnik, and M. Ostendorf, "Data selection with fewer words," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 58–65, 2015.

[130] M. van der Wees, A. Bisazza, and C. Monz, "Dynamic data selection for neural machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 7–11, 2017.

[131] K. Yasuda, R. Zhang, H. Yamamoto, and E. Sumita, "Method of selecting training data to build a compact and efficient translation model.," in *Proceedings of the International Joint Conference on Natural Language Processing*, pp. 655–660, 2008.

[132] H. Daumé III and J. Jagarlamudi, "Domain adaptation for machine translation by mining unseen words," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 407–412, 2011.

[133] P. Banerjee, S. K. Naskar, J. Roturier, A. Way, and J. van Genabith, "Domain adaptation in smt of user-generated forum content guided by oov word reduction: Normalization and/or supplementary data," in *Proceedings of the Annual Meeting of the European Association for Machine Translation*, pp. 169–176, 2012.

[134] P. Banerjee, R. Rubino, J. Roturier, and J. van Genabith, "Quality estimation-guided supplementary data selection for domain adaptation of statistical machine translation," *Machine Translation*, vol. 29, no. 2, pp. 77–100, 2015.

[135] G. Gascó, M.-A. Rocha, G. Sanchis-Trilles, J. Andrés-Ferrer, and F. Casacuberta, "Does more data always yield better translations?," in *Proceedings of the European Chapter of the Association for Computational Linguistics*, pp. 152–161, 2012.

[136] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation.," in *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pp. 678–683, 2013.

[137] B. Chen, R. Kuhn, G. Foster, C. Cherry, and F. Huang, "Bilingual methods for adaptive training data selection for machine translation," *Proceedings of the Conference of the Association for Machine Translation in the Americas*, pp. 93–106, 2016.

[138] B. Chen and F. Huang, "Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data," *Proceedings of the Conference on Computational Natural Language Learning*, pp. 314–323, 2016.

[139] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domains," in *Proceedings of the International Workshop on Spoken Language Translation*, pp. 95–105, 2015.

[140] C. Servan, J. Crego, and J. Senellart, "Domain specialization: a post-training domain adaptation for neural machine translation." *arXiv:1612.06141*, 2016.

[141] A. V. M. Barone, B. Haddow, U. Germann, and R. Sennrich, "Regularization techniques for fine-tuning in neural machine translation." *arXiv:1707.09920*, 2017.

[142] M. Freitag and Y. Al-Onaizan, "Fast domain adaptation for neural machine translation." *arXiv:1612.06897*, 2016.

[143] N. Durrani, H. Sajjad, S. Joty, and A. Abdelali, "A deep fusion model for domain adaptation in phrase-based mt," in *Proceedings of the International Conference on Computational Linguistics*, pp. 3177–3187, 2016.

[144] L. Zhou, W. Hu, J. Zhang, and C. Zong, "Neural system combination for machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 378–384, 2017.

[145] C. Kobus, J. Crego, and J. Senellart, "Domain control for neural machine translation." *arXiv:1612.06140*, 2016.

[146] R. Sennrich, B. Haddow, and A. Birch, "Controlling politeness in neural machine translation via side constraints.," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 35–40, 2016.

[147] W. Chen, E. Matusov, S. Khadivi, and J.-T. Peter, "Guided alignment training for topic-aware neural machine translation." *arXiv:1607.01628*, 2016.

[148] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation." *arXiv:1503.03535*, 2015.

[149] J. Zhang and C. Zong, "Bridging neural machine translation and bilingual dictionaries." *arXiv:1610.07272*, 2016.

[150] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[151] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[152] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[153] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[154] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," *Artificial Neural Networks: Formal Models and Their Applications*, pp. 753–753, 2005.

[155] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 189–196, 1995.

[156] M. D. Zeiler, "Adadelta: an adaptive learning rate method." *arXiv:1212.5701*, 2012.

[157] D. Kingma and J. Ba, "Adam: A method for stochastic optimization." *arXiv:1412.6980*, 2014.

[158] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, *et al.*, "Theano: A python framework for fast computation of mathematical expressions." *arXiv:1605.02688*, 2016.

[159] V. Ambati, S. Vogel, and J. G. Carbonell, "Active learning and crowd-sourcing for machine translation," in *Proceedings of the Conference on Language Resources and Evaluation*, pp. 2169–2174, 2010.

[160] S. Ananthakrishnan, R. Prasad, D. Stallard, and P. Natarajan, "A semi-supervised batch-mode active learning strategy for improved statistical machine translation," in *Proceedings of the Conference on Computational Natural Language Learning*, pp. 126–134, 2010.

[161] A. Bisazza, N. Ruiz, and M. Federico, "Fill-up versus interpolation methods for phrase-based smt adaptation.," in *Proceedings of International Workshop on Spoken Language Translation*, pp. 136–143, 2011.

[162] M. Cettolo, C. Servan, N. Bertoldi, M. Federico, L. Barrault, and H. Schwenk, "Issues in incremental adaptation of statistical mt from human post-edits," in *Proceedings of the Workshop on Post-editing Technology and Practice*, pp. 111–118, 2013.

[163] S. Joty, N. Durrani, H. Sajjad, and A. Abdelali, "Domain adaptation using neural network joint model," *Computer Speech & Language*, vol. 45, pp. 161–179, 2017.

[164] P. Dakwale and C. Monz, "Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data," *Proceedings of the Machine Translation Summit*, pp. 156–169, 2017.

[165] J. Niehues and A. Waibel, "Domain adaptation in statistical machine translation using factored translation models," in *Proceedings of the Conference of the European Association for Machine Translation*, pp. 1–8, 2010.

[166] C. Hui, H. Zhao, Y. Song, and B.-L. Lu, "An empirical study on development set selection strategy for machine translation learning," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 67–71, 2010.

[167] M. Li, Y. Zhao, D. Zhang, and M. Zhou, "Adaptive development data selection for log-linear model in statistical machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 662–670, 2010.

[168] Z. Zheng, Z. He, Y. Meng, and H. Yu, "Domain adaptation for statistical machine translation in development corpus selection," in *Proceedings of the International Universal Communication Symposium*, pp. 2–7, 2010.

[169] L. Liu, H. Cao, T. Watanabe, T. Zhao, M. Yu, and C. Zhu, "Locally training the log-linear model for smt," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 402–411, 2012.

[170] X. Song, L. Specia, and T. Cohn, "Data selection for discriminative training in statistical machine translation," in *Proceedings of the Conference of the European Association for Machine Translation*, pp. 45–53, 2014.

[171] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 20-29, pp. 707–710, 1966.

[172] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 129–136, 2011.

[173] J. Allan, C. Wade, and A. Bolivar, "Retrieval and novelty detection at the sentence level," in *Proceedings of the Annual International Conference on Research and Development in Information Retrieval*, pp. 314–321, 2003.

[174] M. Chinea-Rios, G. Sanchis-Trilles, and F. Casacuberta, "Bilingual data selection using a continuous vector-space representation," in *Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition*, pp. 95–106, 2016.

[175] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting tf-idf term weights as making relevance decisions," *ACM Transactions on Information Systems*, vol. 26, pp. 13–30, 2008.

[176] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[177] B. Haddow and P. Koehn, "Analysing the effect of out-of-domain data on smt systems," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 422–432, 2012.

[178] A. Irvine, J. Morgan, M. Carpuat, H. Daumé III, and D. Munteanu, "Measuring machine translation errors in new domains," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 429–440, 2013.

[179] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 11–19, 2015.

[180] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation." *arXiv:1604.02201*, 2016.

[181] J. M. C. Dakun Zhang, Jungi Kim and J. Senellart, "Boosting neural machine translation." *arXiv:1612.06138*, 2016.

[182] R. Wang, M. Utiyama, A. Finch, L. Liu, K. Chen, and E. Sumita, "Sentence selection and weighting for neural machine translation domain adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1727–1741, 2018.

[183] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725, 2016.

[184] D. Britz, A. Goldie, T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures." *arXiv:1703.03906*, 2017.

[185] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization." *arXiv:1607.06450*, 2016.

[186] A. Graves, "Practical variational inference for neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2348–2356, 2011.

[187] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks." *arXiv:1211.5063*, 2012.

[188] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." *arXiv:1609.08144*, 2016.

[189] M. Chinea-Rios, A. Peris, and F. Casacuberta, "Are automatic metrics robust and reliable in specific machine translation tasks?," in *Proceedings of the Annual Conference of the European Association for Machine Translation*, pp. 89–98, 2018.

[190] P. Martínez-Gómez, G. Sanchis-Trilles, and F. Casacuberta, "Online adaptation strategies for statistical machine translation in post-editing scenarios," *Pattern Recognition*, vol. 45, no. 9, pp. 3193–3203, 2012.

[191] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[192] B. Chen and C. Cherry, "A systematic comparison of smoothing techniques for sentence-level bleu," in *Proceedings of the Workshop on Statistical Machine Translation*, pp. 362–367, 2014.