UNIVERSITAT POLITÈCNICA DE VALÈNCIA

MIARFID Official Master's Degree in Artificial Intelligence, Pattern Recognition and Digital Imaging

Departamento de Sistemas Informáticos y Computación

Universitat Politècnica de València

On The Use of Monolingual Corpus for Training Neural Machine Translation Systems

MASTER'S THESIS

Master's Degree in Artificial Intelligence, Pattern Recognition and Digital Imaging.

Author: Mónica Lorena Castellanos Pellecer

Tutor: Francisco Casacuberta Nolla

Course 2019-2020

Abstract

The quality of machine translation produced by state-of-the-art models is already quite high and often requires only minor corrections from professional human translators. This is especially true for high-resource language pairs like English-German and English-French. So, the main focus of recent research studies in machine translation was on improving system performance for low-resource language pairs, where we have access to large monolingual corpora in each language but do not have sufficiently large parallel corpora.

The most successful approach to date is the proposal of [1], who use monolingual target texts to generate artificial parallel data via backward translation (BT). This technique has since proven effective in many subsequent studies. It is however very computationally costly, typically requiring translating large sets of data.

In this master thesis, a methodology that combines backward translation and different ratios of the real and synthetic (pseudo bilingual) corpora is proposed in order to, given a MT system, increase the translation quality of that system. We will compare the results with a baseline obtained by using all the bilingual corpora.

This methodology is tested on different scenarios where we divided the corpora in two equal sizes (Bilingual and Monolingual), then each part was divided in different ratios to use backward translation and just mixing real corpus and pseudo bilingual corpus. Finally, we obtain very encouraging results.

Keywords: neural machine translation; machine translation with monolingual corpus; back translation.

Resumen

La calidad de la traducción automática producida por los modelos de última generación ya es bastante alta y a menudo sólo requiere correcciones menores por parte de traductores humanos profesionales. Esto es especialmente cierto para pares de idiomas de alto recurso como el inglés-alemán y el inglés-francés. Por lo tanto, el enfoque principal de los recientes estudios de investigación en traducción automática fue mejorar el rendimiento del sistema para los pares de idiomas de bajos recursos, donde tenemos acceso a grandes corpus monolingües en cada idioma, pero no tenemos corpus paralelos suficientemente grandes.

El enfoque más exitoso hasta la fecha es la propuesta de [1], que utiliza textos de destino monolingües para generar datos paralelos artificiales a través de la traducción inversa (BT). Desde entonces, esta técnica ha demostrado su eficacia en muchos estudios posteriores. Sin embargo, es muy costosa desde el punto de vista informático, ya que normalmente requiere la traducción de grandes conjuntos de datos.

En esta tesis de maestría se propone una metodología que combina la traducción inversa y diferentes proporciones de los corpus reales y sintéticos (pseudo bilingües) con el fin de, dado un sistema de MT, aumentar la calidad de la traducción de ese sistema. Compararemos los resultados con una línea de base obtenida utilizando todos los corpus bilingües.

Esta metodología se prueba en diferentes escenarios en los que dividimos los corpúsculos en dos tamaños iguales (una bilingüe y otra monolingüe), y luego cada parte se dividió en diferentes proporciones para utilizar la traducción inversa y mezclar sólo el corpus real y el corpus pseudo bilingüe. Finalmente, obtenemos resultados muy esperanzadores.

Palabras clave: traducción automática neuronal; traducción automática con corpus monolingües; traducción inversa.

Acknowledgments

I would like to thank my family for all their unconditional love.

My most sincere gratitude to Dr. Francisco Casacuberta, for supervising this thesis and giving me this opportunity. Also, I have to thank Miguel Domingo, whose help and advices were crucial for the development of this thesis.

I gratefully acknowledge the support of NVIDIA Corporation with the donation of a GPU used for this research.

Thanks!

Contents

List of Figures

List of Tables

CHAPTER 1
Introduction and Objectives

1.1 Introduction

Human language, unique among living beings for its level of complexity and sophistication, could be born closely to the origins of modern human behavior, but there is little agreement about the implications and directionality of this connection. Today, there are various hypotheses about how, why, when, and where language might have merged [2].

Despite this, there is barely more agreement today than when Charles Darwin's theory of evolution by natural selection provoked speculation on the topic [3]. However, since the early 1990s a several group of scientists (linguists, psychologists, archaeologists, anthropologists, etc.) have broach with new methods these topics that is consider one of the hardest problems in science [4].

One of the natural human behaviors is to socialize and for that we use the language. We live in a multilingual world where communication between different languages becomes a great challenge. From this obviously necessity, Machine Translation (MT) was born.

The statistical models have dominated this field during the last decades. These models rely on the use of huge bilingual corpora to be trained. In the last years, neural networks have established as the state of the art of machine translation replacing other classical approximations and thanks to these advances in MT we can have many different software and toolkits to study even more the deep world of MT.

While monolingual data has been shown to be useful in improving bilingual neural machine translation (NMT), effectively and efficiently leveraging monolingual data for Multilingual NMT (MNMT) systems is a less explored area [5].

Utilizing monolingual data has been widely explored in various NMT and natural language processing (NLP) applications. Back translation (BT) [5], which leverages a target-to-source model to translate the target-side monolingual data into source language and generate pseudo bitext, has been one of the most effective approaches in NMT. However, well trained NMT models are required to generate back translations for each language pair, it is computationally expensive to scale in the multi- lingual setup [5].

In this master thesis, we propose a methodology to build a translator by using a proportion of bilingual and monolingual corpus to the training process and use Back

translation (BT) or inverse translator to generate a pseudo corpus in the middle of the process. Then we will compare the translator quality with the baseline that will be a translator trained with the same hyperparameters but using a full bilingual corpus.

1.2 Objectives

Neural Machine Translation (NMT) has obtained state-of-the art performance for several language pairs, while only using parallel data for training.

The new generation of NMT systems is known to be extremely data hungry [6]. Yet, most existing NMT training pipelines fail to fully take advantage of the very large volume of monolingual source and/or parallel data that is often available [7].

So, the main focus of recent research studies in machine translation was on improving system performance for low-resource language pairs, where we have access to large monolingual corpora in each language but do not have sufficiently large parallel corpora.

In this master thesis we described a methodology following some techniques of machine translation with monolingual corpus to be able to compare if it's possible to use a pseudo parallel corpus for machine translation with monolingual corpus.

We still don't know the impact of the synthetic corpus size and in several researches like in [8], they mention that this is not very intuitive, so we decided to lead several experiments with different corpus and pseudo bilingual corpus size, using back-translation and mix them.

We are going to guide this master thesis through all the following chapters, first to have the main idea and understand the machine translation with monolingual corpus, then we will describe the task of the experiments, corpora, software, metrics and the experimental setup following by the results obtained and finally we will present our conclusions and future work.

CHAPTER 2
Machine Translation

2.1 Machine Translation

Social phenomena such as the globalization and technological development have dramatically increased the need of translation information from one language to another. This necessity can be found in different fields including political institutions, industry, education or entertainment. Translation from one language to another by using a computer is the main objective of MT.

The principal strategies that have been applied to MT can be classified as follows [9]:

- In line with the input type: text or speech
- In line with the type of application which uses the translation: applications that translate the input into a database query; applications that produce an approximated translation of the input for its correction in a post-edition stage by the user; applications that interactively generate the output in collaboration with the user; or fully automated translation systems.
- In line with the translation technology: rule-based or corpus-based systems.

2.1.1 Rule-Based Systems

Rule-Based Machine Translation (RBMT) was one of the first approaches used in MT [10]. RBMT use a set of translation rules created by human translators to generate their output. These rules are what determines how to translate from one language to another.

These systems execute two different steps to generate their translations: the analysis step and the generation step. Depending on the mentioned steps, RBMT can be classified in three stages: direct approach, the transfer approach and the interlingual approach.

2.1.2 Corpus-Based Systems

Corpus-based system use sets of translation examples (also called corpus or parallel texts) from one language to another. These examples are used to infer the translation of the source text. Once a corps-based system has been implemented, the software

can be quickly adapted for its use with different language pairs or different domains, as opposed to rule-based systems, which are specific for a given language pair.

Corpus-based system can be classified in three main groups [9]:

- **Example-Base Machine Translation (EBMT) systems:** these systems use a set of translations examples as its main knowledge base. Translation process is generated through two steps: first, a set of hypotheses similar to the source text are extracted from the corpus (comparison); and second, the hypotheses are recombined to generate the final translation of the source text (recombination).
- **Statistical Machine Translation (SMT) systems:** these systems base their translations on statistical models and other models from information theory. They require a great amount of parallel texts containing relevant information for the translation process. These texts are used to estimate the parameters of the models mentioned before, which are used to infer the translation of a new source text.
- **Other corpus-based systems:** there are other alternatives to implement corpus-based systems, such as the finite state approach, which applies the mathematical tools provided by the automata theory; or the context-free grammar approach, which applies context-free grammars to MT.

2.2 Statistical Machine Translation

For decades Statistical Machine Translation (SMT) has been the state-of-the-art. SMT approaches the MT problems, as its name suggests, from a statistical point of view. Subsequently, statistical models are involved in the translation process and these models estimates their parameters from the parallel texts of the corpora.

Hence, if the needed corpora are available, SMT systems can work to translate many different languages.

Explaining this in a formal way, given a sentence $\boldsymbol{x}$ in the source language, the work in MT is to find its corresponding translation $\boldsymbol{y}$ in the target language.

Formalizing this problem in SMT [11]:

$$\hat{y} = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y}|\boldsymbol{x}) \tag{2.1}$$

Applying the famous Bayes' theorem, we can also have the following equation:

$$\hat{y} = \arg\max_{\boldsymbol{y}} \Pr(\boldsymbol{y}) \cdot Pr(\boldsymbol{x}|\boldsymbol{y}) \tag{2.2}$$

This last step is known as fundamental equation of machine translation [11]. The term $\Pr(\boldsymbol{y})$ represents the well-formedness of $\boldsymbol{y}$, and is usually called the language model probability and the term $Pr(\boldsymbol{x}|\boldsymbol{y})$ corresponds to the translation model, which represents the relation between the source sentences and its translation.

Putting all this in practice, we often combine all these models into a log-linear model for $Pr(\boldsymbol{x}|\boldsymbol{y})$ [12]:

$$\hat{y} = \arg\max_{\boldsymbol{y}}\{\sum_{n=1}^{N} \lambda_n \cdot log\ (f_n(\boldsymbol{y}, \boldsymbol{x}))\} \tag{2.3}$$

Where $f_n(\boldsymbol{y}, \boldsymbol{x})$ can be any model that represents an important feature for the translation; N is the number of models (or features); and λ_n are the weights of the log-linear combination.

The log-linear models can be represented in different way, the most popular are those who include phrase-based models [13] [14] which basic idea is the segmentation of the source sentence into phrases; the translation of those source phrases into target phrases; and the reordering of those translated phases in order to compose the target sentence.

Therefore, there are three main computational challenges of SMT [15]:

1. Estimating the language model probability.
2. Estimating the translation model probability.
3. Finding an efficient and effective global search method.

In order to build a translation system based on Bayes' rule, these computational challenges are represented as different modules. Additionally, a preprocess and a postprocess phase for the sentences are necessary. This is explained in the following diagram.

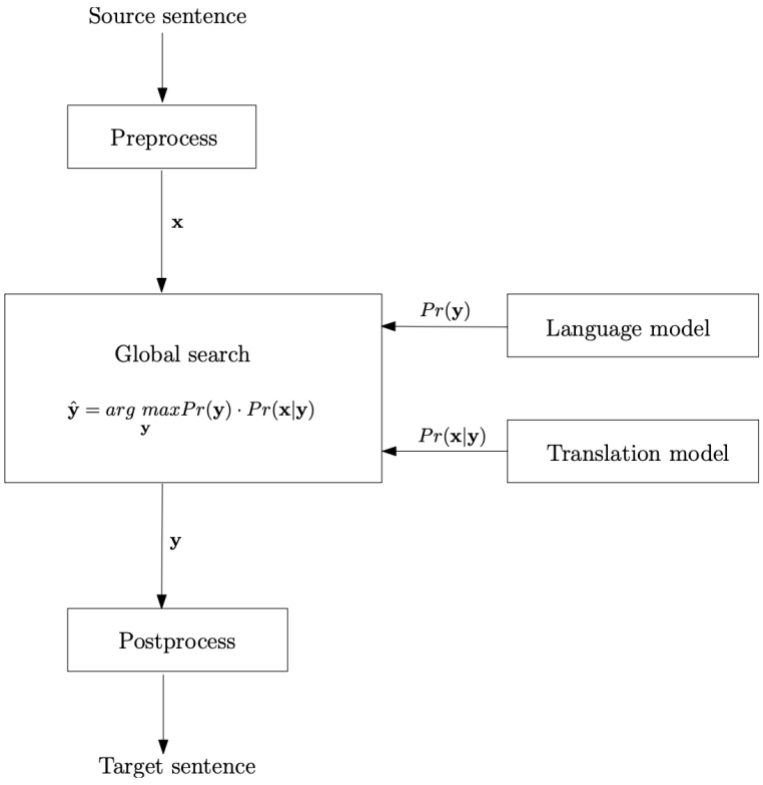

Figure 2.1: Flow diagram of the translation process based on Bayes' rule. Figure extracted from [14].

2.3 Neural Machine Translation

Neural Machine Translation (NMT) has emerged to constitute itself as the state-of-the art in automatic translation as a new approach in MT problems. The translation process is carried out by artificial neural networks or simply neural networks (NN).

Neural networks are models whose basic unit is the neuron, which performs a linear combination of its inputs and later applies a non-linear function to the output. These simple units are used to build complex networks that have a wide range of applications.

The main idea of Neural Machine Translation (NMT) is represented by the following equation:

$$\hat{s} = arg\max_{s} \prod_{i=1}^{|s|} pr_{\theta}\left(s_i \middle| s_1^{i-1}, c(e)\right) \tag{2.4}$$

Where s_i is the current translated word, which is generated from the s_1^{i-1} words that were previously translated with a type of representation named by the function c of the origin sentence e, and the parameters of the model θ. Explaining this in a different way, the model is often trained to predict the next word s_i, given c and all words $s_1, \dots, s_{i-1}$.

Unlike SMT system, NMT fit a parameterized model to maximize the conditional probability of sentence pairs using a parallel training corpus. Once the conditional distribution is learned by a translation model, given a source sentence a corresponding translation can be generated by searching for the sentence that maximizes the conditional probability.

Most of the proposed neural machine translation models belong to a family of encoder-decoders [16]; [17] with an encoder and a decoder for each language, or involve a language-specific encoder applied to each sentence whose outputs are then compared [18]. An encoder neural network reads and encodes a source sentence into a fixed-length vector. A decoder then outputs a translation from the encoded vector. The whole encoder–decoder system, which consists of the encoder and the decoder for a language pair, is jointly trained to maximize the probability of a correct translation given a source sentence.

A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus. [19] showed that indeed the performance of a basic encoder–decoder deteriorates rapidly as the length of an input sentence increases.

2.3.1 RNN Encoder-Decoder

The RNN Encoder-Decoder (RED) have similarities with Auto-Encoder (AE) that generate the output from the inputs. The RED used to predict the next sequences in general [20].

In the RNN Encoder-Decoder (RED), each RNN cell can be selectively used among vanilla RNN [21], Long-short Term Memory (LSTM) [22] and Gated Recurrent Unit (GRU) [23]. The vanilla RNN has vanishing gradient problem when the length of the sequence becomes long. However, LSTM and GRU solved vanishing gradient problem.

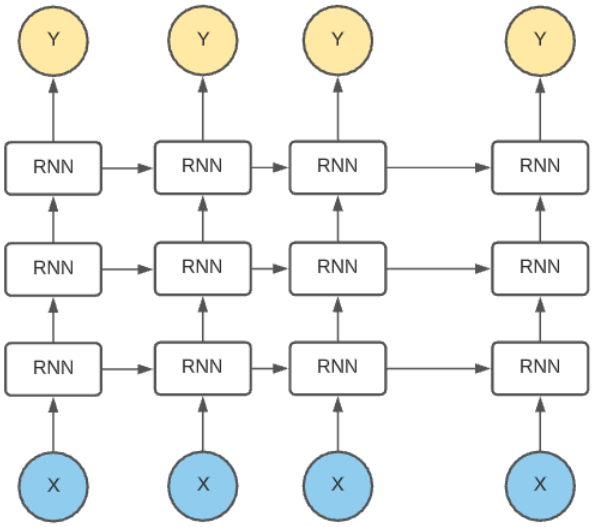

Figure 2.2: The RNN Encoder-Decoder architecture. It generates sequential output from sequential input.

The encoder-decoder system is trained to maximize the conditional log-likelihood over a set of bilingual phrases $C = \{(c, y_1), \dots, (c_N, y_N)\}$. [24] look for the optimal set of parameters that maximizes that probability over the training set:

$$\hat{\theta} = \arg\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} log p_\theta(y_n | c_n) \tag{2.5}$$

Where $c_n = c_1, \dots, c_J$ is an input statement of size J, $y_n = y_1, \dots, y_I$ is an output statement of size I and finally θ is the set of parameter. The input of these networks is the source language, and its output is the target language which has the real translation.

Encoder

The encoder is usually composed of a recurrent neural network with hidden layers of the type "Long Short Term Memory" (LSTM) [22]. The encoder projects a representation of its input in a fixed size vector, while the decoder is fed with this

representation to produce the translation of the sentence in the target language [19], as shown in the next figure:

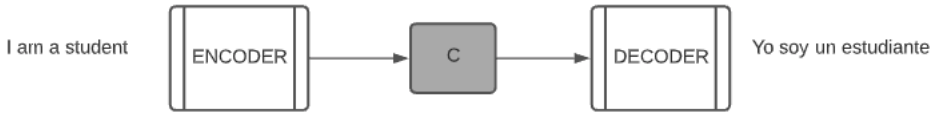

Figure 2.2: Encoder – Decoder configuration.

The representation of words and phrases is one difference that characterizes neural automatic translation, which are projected in numerical vectors.

The encoder is formally modeled in the following way [19] [16].

$$h_j = f(x_j, h_{j-1}) \tag{2.6}$$

and

$$c = q(h_1, \dots, h_j) \tag{2.7}$$

Where x_j is an input sequence to the model, $h_j \in R_n$ is the hidden state in time j, and c is a vector generated based on the sequence of hidden states, while f and q are a non-linear function.

Decoder

The decoder is trained to generate the output statement $y_1^I = y_1, \dots, y_I$ given the predicted words above and the context vector c. From a probabilistic perspective the decoder is defined as:

$$p(y) = \prod_{i=1}^{I} p(y_i | \{y_1, \dots, y_{i-1}\}, c) \tag{2.8}$$

Where $y = (y_1, \dots, y_I)$. Using a RNN such as in this case, the probability is modeled as follows:

$$p(y_i | \{y_1, \dots, y_{i-1}\}, c) = g\big(y_{i-1}, s_i, c\big) \tag{2.9}$$

Where g is a non-linear function and s_i is the hidden state of the RNN.

2.3.2 Attention Model

The use of a fixed size vector in the Encoder-Decoder model produces a bottleneck, since the vector being of fixed size cannot capture the full context to the sentence when it is of an extensive size [19]. However, this problem is solved by creating a variable context vector, which captures the context and relationships of the statements to be translated. Formally, this approach is modeled as follows:

$$p(y_i|\{y_i, \dots, y_{i-1}\}, c_i) = g(y_{i-1}, s_i, c_i) \tag{2.10}$$

Where g is a non-linear function, s_i is the hidden state of the RNN and the vector c_i is the weighted sum of the encoder's outgoing annotation sequence, which is represented as:

$$c_i = \sum_{j=1}^{J} \alpha_{ij} h_j \tag{2.11}$$

Where α_{ij} is estimated by a softmax function for each h_j. This estimation is done by the following equation:

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{J} \exp(e_{ik})} \tag{2.12}$$

Where $e_{ij} = a(s_{i-1}, h_j)$ is a score given by an alignment model, which measures how well the j inputs positions match the j outputs positions.

This alignment model receives the previous state of the s_{i-1} decoder, and the notation of the h_j source statement. Formally, this model is described as follows:

$$A(s_{i-1}, h_j) = v_a^{\mathrm{T}} \tanh(W_a S_{i-1} + U_a h_j) \tag{2.13}$$

Where v_a, W_a and U_a, are the trainable weights of the attention model. This allows to obtain a representation of the entry sentence based on the training of these weights.

There are other approximations in the literature for calculating the attentional model, such as the attentional point product [25], which is formally written as follows:

$$A(s_{i-1}, h_j) = s_{i-1}^{\mathrm{T}} h_j \tag{2.14}$$

This approach has an advantage because it does not have parameters as in equation 2.12. This advantage lies in the simple dot product between the query and key projection vectors.

2.3.3 NMT: Sub-word translation

Extensive research exists for some languages [26], [27] about word-level translation. In [24] they end up using suboptimal approaches for preprocessing and tokenizing the dataset. This results in having "fly", "flies", "flew", "flying" as different entries in the vocabulary and treated as if they were completely different words which consequence in:

- Machine Translation systems have problems when generalizing into new words for the reason that it ignores the word structure.
- Difficult modeling morphological variants in an efficient way.

These issues can be addressed to a certain extend by using characters or other sub-word units. Different NMT approximations might be classified as [24]:

- **Sub-word-level Encoder:** It applies character or sub-word treatment to the source language; the encoder is able to interpret unseen words.
- **Sub-word-level Decoder:** It applies character or sub-word treatment to the target language; the decoder is able to generate new words.
- **Complete Sub-word-level:** The source and target language are handled at character or sub-word level. This is the closest approximation to an open vocabulary system.

2.3.4 Byte Pair Encoding (BPE)

Byte Pair Encoding (BPE) is a popular technique for working with sub-word units. BPE deals with rare and unknown words by encoding them as a sequence of sub-word units. This is based on the intuition that some word classes are translatable via smaller units: compounds, loanwords or cognates.

BPE was used by [28] with one goal: to propose a simple and effective way to achieve open vocabulary. To do so they employed a data compression algorithm on words: BPE [29]. BPE is an iterative algorithm which at each time step takes the two most frequent symbols and merges them, this way the algorithm learns the substitution rules from the datasets separated in characters. Pairs that cross word boundaries are not considered. BPE is a statistical approach, this results in frequent character n-grams (or words) merged into a single symbol. On the other hand, rare n-grams result in short symbols or single characters. Here we present a simple sequence of steps to apply BPE over a dataset:

1. Text is tokenized in characters.
2. Set desired number of merges. The only hyper-parameter of the algorithm.
3. Search for the most frequent pair of symbols.
4. Merge the pair as a new single symbol.
5. Go to step 3 until the number of merge operations is reached.

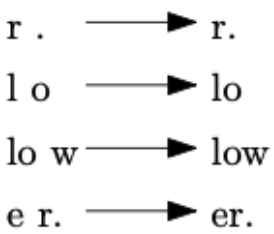

Figure 2.4: BPE merge operations learned from dictionary {'low', 'lowest', 'newer', 'wider'}. Example borrowed from [28]

CHAPTER 3
Machine Translation with monolingual corpus

3.1 Machine Translation with monolingual corpus.

Neural Machine Translation (NMT) has obtained state-of-the art performance for several language pairs, while only using parallel data for training.

The new generation of Neural Machine Translation (NMT) systems is known to be extremely data hungry [6]. Yet, most existing NMT training pipelines fail to fully take advantage of the very large volume of monolingual source and/or parallel data that is often available [7].

The most successful approach to date is the proposal of [1], who use monolingual target texts to generate artificial parallel data via backward translation (BT). This technique has since proven effective in many subsequent studies. It is however very computationally costly, typically requiring translating large sets of data. Determining the "right" amount (and quality) of BT data is another open issue, but we observe that experiments reported in the literature only use a subset of the available monolingual resources. This suggests that standard implementations for BT might be sub-optimal.

3.1.1 Forward and Back-Translation

Given the translation task L1→L2, where a large scale monolingual data are available in L2, back translation refers to training a translation model L2→L1 and using it to translate the L2 data to create a synthetic data that can be added to the true bilingual data to train a L1→L2 model. Back translation was first explored for statistical machine translation (SMT) but was found to be much more effective for Neural MT, particularly in a low resource scenario.

To use forward translation, the monolingual data should be available in L1, which is translated using a model L1→L2, and added to the true bilingual corpus for retraining the L1→L2 model (aka self-training). Self-training with forward translation was also pioneered in SMT, but it has shown that NMT can also benefit using the same. Compared to back translation, error and biases are intuitively more problematic when using forward-translation as they directly affect the encoder training [30].

3.1.2 Effect of synthetic data

In [31] used all the available test-data in the news domain for French(FR) – English(EN), and split them based on the source language (natural vs human translation). In the experiments (FR⟶EN), they reported that the back-translation had a relative gain of 6.8 BLEU (see section 4.5.1 for BLEU definition) points for the portion of the test-sets with reverse translation whereas forward translation improves them by only 1.00 BLEU. However, on the test-sets that were originally in source language, forward translation brought an improvement of 2.00 BLEU points, whereas back-translation has suffered an average loss of 1.00 BLEU.

With the above results, we can conclude that back-translation is more effective than forward translation in the somewhat artificial setting where the input to the translation system is itself a human translation, and the original text is used as reference. In the more natural setting where the input is native text, and the reference is a human translation, forward translation can perform better in terms of BLEU [31].

The research done in [8] they demonstrate improvements in neural machine translation quality in both high and low resourced scenarios, including the best reported BLEU scores for the WMT 2017 German <-> English tasks.

The following figure shows the idea when an NMT system is trained in the reverse translation direction (target to source) and is then used to translate target-side monolingual data back into the source language (in the backward direction, hence the name backtranslation). The resulting sentence pairs constitute a synthetic parallel corpus that can be added to the existing training data to learn a source-to-target model.

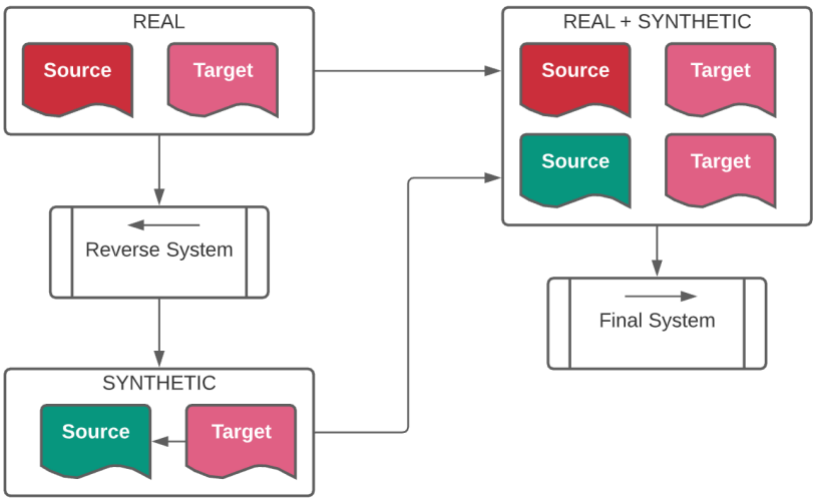

Figure 3.1: Creating a synthetic parallel corpus through back-translation. First, a system in the reverse direction is trained and then used to translate monolingual data form the target side backward into the source side, to be used in the final system.

The researches in [8] also presented the idea of iterative back-translation: "If we can build a better system with the back-translated data, then we can continue repeating this process. Use this better system to back translate the data and use this data in order to build an even better system". See the figure 3.2 and figure 3.3 for an illustration and detailed algorithm of this iterated back-translation process.

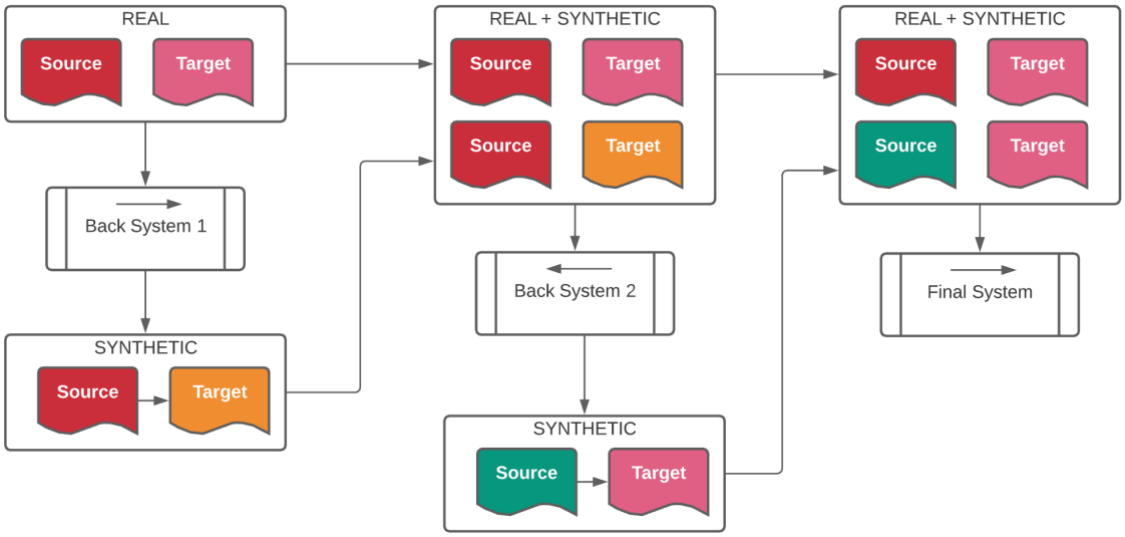

Figure 3.2: Iterative Back-Translation: After training a system with back-translated data (back system 2 above), it is used to create a synthetic parallel corpus for the final system.

Algorithm 1 Iterative Back-Translation

Input: parallel data D^p, monolingual source, D^s, and target D^t text

Let $T_{\leftarrow} = D^p$
repeat
  Train target-to-source model $\Theta_{\leftarrow}$ on $T_{\leftarrow}$
  Use $\Theta_{\leftarrow}$ to create $S = \{(\hat{s}, t)\}$, for $t \in D^t$
  Let $T_{\rightarrow} = D^p \cup S$
  Train source-to-target model $\Theta_{\rightarrow}$ on $T_{\rightarrow}$
  Use $\Theta_{\rightarrow}$ to create $S' = \{(s, \hat{t})\}$, for $s \in D^s$
  Let $T_{\leftarrow} = D^p \cup S'$
until convergence condition reached

Output: newly-updated models $\Theta_{\leftarrow}$ and $\Theta_{\rightarrow}$

Figure 3.3: Iterative Back-Translation Algorithm: After training a system with back-translated data, it is used to create a synthetic parallel corpus for the final system. We repeat this until convergence condition reached or as many times we want. Figure extracted from [8].

In the paper mentioned above [8] they vary the amount of synthetic data to compare the results. As pointed out by [31] the balance between the real and synthetic parallel data matters. However, there is no obvious evidence about the effect of the sample size. In [8] they decided to use the following ratios: (real) : (synthetic), 1:1,

1:2, 1:3 and their result show that more synthetic parallel data seems to be useful (though not obvious), e.g., gains from 16.7 to 16.9 in English to Farsi and gain from 22.1 to 22.3 in Farsi to English (using BLEU metrics).

We know that back translation-data augmentation by translating target monolingual data – is a crucial component in modern NMT. In [32] they reformulate back-translation in the scope of cross entropy optimization of an NMT model, clarifying its underlying mathematical assumptions and approximations beyond its heuristic usage. Their formulation covers broader synthetic data generation schemes, including sampling from a target-to-source NMT model.

In this study [32], they compare different monolingual corpus sizes for the German to English task on three different test sets by doing beam search and additional sampling-based methods. They identify that the search method plays an important role, as it is responsible for offsetting the shortcomings of the generator model. Specifically, label smoothing and probability smearing issues cause sampling-based methods to generate unnatural sentences. In terms of translation quality sampling from 50-best lists outperforms beam search, albeit at a higher computation cost. Restricted sampling or the disabling of label smoothing for the generator model are shown to be cost-effective ways of improving upon the unrestricted sampling approach of [33].

Even if there is some works done about back translation and the use of monolingual corpus for having advantages and improve NMT there is still nothing standardized to follow and let us advance for doing more researches. As we noticed in [8] and [32] they got results and affirming that use of back translation can lead us to obtain good quality translation comparing with some baseline.

In [8] they use some iteration of back translation to build a synthetic/pseudo corpus trying some combination of the synthetic corpus with the real corpus. Nonetheless, it is still not obvious evidence and for nothing intuitive to know the effect of the sample size. In [32] they wanted to generalize the use of back-translation by applying different techniques and probabilities to choose wisely and with mathematics background the sample size, but the results they got, just let open to keep investigating and doing more experiments using their results as a guide.

3.1.3 State-Of-The-Art

The quality of machine translation produced by state-of-the-art models is already quite high and often requires only minor corrections from professional human translators. This is especially true for high-resource language pairs like English-German and English-French. So, the main focus of recent research studies in machine translation was on improving system performance for low-resource language pairs, where we have access to large monolingual corpora in each language but do not have sufficiently large parallel corpora.

Facebook AI researchers [34] seem to lead in this research area and have introduced several interesting solutions for low-resource machine translation during the last year. This includes augmenting the training data with back-translation, learning

joint multilingual sentence representations, as well as extending BERT to a cross-lingual setting.

In [34] they investigate how to learn to translate when having access to only large monolingual corpora in each language. They propose two model variants, a neural and a phrase-based model. Both versions leverage a careful initialization of the parameters, the denoising effect of language models and automatic generation of parallel data by iterative back-translation. These models are significantly better than methods from the literature, while being simpler and having fewer hyper-parameters. On the widely used WMT'14 English-French and WMT'16 German-English benchmarks, their models respectively obtain 28.1 and 25.2 BLEU points without using a single parallel sentence, outperforming the state of the art by more than 11 BLEU points. On low-resource languages like English-Urdu and English-Romanian, their methods achieve even better results than semi-supervised and supervised approaches leveraging the paucity of available bitexts.

The core idea of the paper [34] was:

Unsupervised MT can be accomplished with:

- Suitable initialization of the translation models (i.e., byte-pair encodings).
- Training language models in both source and target languages for improving the quality of translation models (e.g., performing local substitutions, word reordering).
- Iterative back-translation for automatic generation of parallel data.

There are two model variants, neural and phrase-based:

- Neural machine translation has an additional important property – sharing of internal representations across languages.
- Phrase-based machine translation outperforms neural models on low-resource language pairs, is easy to interpret and fast to train.

In the paper [35] researchers from the University of Hong Kong and New York University use a model-agnostic meta-learning algorithm (MAML) to solve the problem of low-resource machine translation. In particular, they suggest using many high-resource language pairs to find the initial parameters of the model. This initialization allows then to train a new language model on a low-resource language pair using only a few steps of learning. For example, the model initialized using 18 high-resource language pairs, was able to achieve the BLEU score of 22.04 on the new language pair by seeing only around 600 parallel sentences.

We can summarize the main objectives of the work done in [35] as:

- The paper introduces a new meta-learning method, MetaNMT, which assumes using many high-resource language pairs to find good initial parameters and then training a new translation model on a low-resource language starting from the found initial parameters.
- Meta-learning can be applied to low-resource machine translation only if the input and output spaces are shared across all the source and target tasks. However, this is generally not the case since different languages have different vocabularies. To tackle this issue, the researchers dynamically build a vocabulary specific to each language using a key-value memory network.

This last paper leads us to the following future research areas:

- Meta-learning for semi-supervised NMT or learning to learn from monolingual corpora.
- Multi-modal meta-learning, when multiple meta-models are learned, and a new language can freely choose a model to adapt from.

MetaNMT can be used to improve the results of machine translation for language pairs where the available parallel corpora are extremely small.

In the following state-of-the-art [33], the Facebook AI research team investigates back translation for neural machine translation at a large scale. In fact, they augment the parallel training corpus with hundreds of millions of back-translated sentences. A comprehensive analysis of different methods to generate synthetic source sentences shows that synthetic data based on sampling and noised beam search provides the strongest training signal. The experiments demonstrate that Big Transformer architecture combined with back translation achieves state-of-the-art results on WMT'14 English-French[1] and WMT'14 English-German[2] datasets with 45.6 BLEU and 35 BLEU respectively.

This is yet another research paper from the Facebook AI research team [36], in a multilingual domain. In this paper, the researchers introduce a new architecture that learns joint multilingual sentence representations. The system is based on a single language agnostic BiLSTM encoder with a shared vocabulary for 93 languages. The suggested approach establishes a new state-of-the-art for most of the languages on several multilingual tasks including zero-shot cross-lingual transfer, cross-lingual document classification, and bitext mining.

In this research they achieve the following:

- Establishing a new state of the art on zero-shot cross-lingual natural language inference for all languages but Spanish, and thus, outperforming the multilingual BERT model in a zero-shot transfer.
- Getting also state-of-the-art results for most languages in:
- cross-lingual document classification (state of the art for 5 of the 7 language transfers).
- Bitext mining (best result for 3 out of 4 language pairs).
- Introducing a new test set of aligned sentences in 122 languages.

The future research areas that this team suggest are:

- Exploring alternative architectures for the encoder, such as for example, replacing BiLSTM with the Transformer.
- Exploiting monolingual training data in addition to parallel corpora using pre-trained word embeddings, back-translation or other strategies from unsupervised machine translation.

1 https://paperswithcode.com/sota/machine-translation-on-wmt2014-english-french

2 https://paperswithcode.com/sota/machine-translation-on-wmt2014-english-german

- Replacing language-specific tokenization and BPE segmentation with a language agnostic approach[3].

3.1.4 Motivation

By reviewing the papers mentioned above and other work on monolingual data in machine translation, we realized that we still have a big world to discover and study. Previous experiments show that we can lead and build a good quality translator by using monolingual data.

We just have results of techniques by using a certain corpus we can implement and try to build our own methodology and use an available corpus and compare the quality of the translator with a base line.

As [8] got a good results in practice by using back translation and iterative back translation, we will use it in our own methodology, this is equivalent to say that while using back translation we inject some kind of noise to our system and in that way we build a more robust translator but there are also cases that this doesn't improve the results and in other works they just mixed bilingual and monolingual corpora without using back translation and they got acceptable results.

We lead our master thesis by doing a lot of experiments using one single corpus to translate from English to Spanish, in the chapter five, we will explain with more details the corpora, the software and all the hyperparameters used. We first break into two equal parts the whole corpora to use one half as a bilingual parallel corpus and the other half as a monolingual parallel corpus. Each of the division of the corpus we will made subdivision to test the experiment with different ratio of corpus size.

Using the bilingual side, we will train a translator from English to Spanish and save that result. Then using this bilingual corpus, we will use back translation or inverse translation from Spanish to English to build a pseudo-bilingual corpus (aka synthetic corpus). Then we will make some variants to test different combinations of the corpus size to build a translator four and saving the setup for the best results, then we will use them to build a pseudo-corpus T3 (joining and shuffle the pseudo-bilingual corpus and the bilingual corpus) to make a translator two. Finally, we will build another pseudo corpus by mixing and shuffling the bilingual corpus and monolingual corpus by testing it with a translator three with the different corpus sizes. The following figure explain in a general way the main idea of the proposed methodology.

3 https://github.com/google/sentencepiece

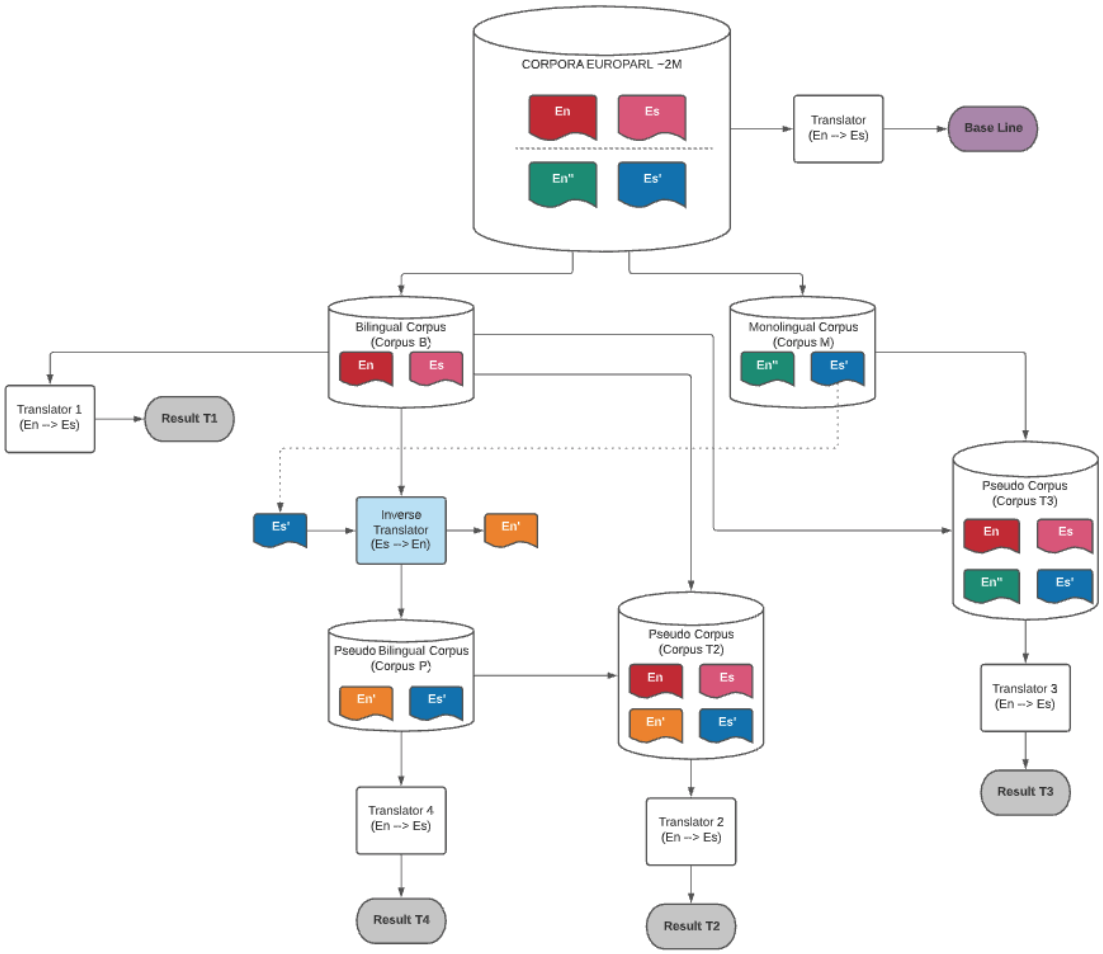

Figure 3.4: This is the general main idea of the methodology used in this master thesis where **En** means for English, **Es** for Spanish; T1, T2, T3, T4 means for the result obtained with translator 1, 2, 3 and 4. Pay special attention with **the first apostrophe and double apostrophe** in each subdivision of the corpus to indicate the source and target side.

In section 4.6 we will explain with more details each module of the above figure.

At the end, we will be able to compare each result obtained in every phase and compare if we have significantly improved by using back translator or not while translating with real and synthetic corpus.

CHAPTER 4
Experimental Framework and results

4.1 Experimental Framework and Results

We will lead this chapter by introducing first our experimental framework for the task that will be described. We will give details about the corpora, software, hyperparameters and metrics used. Then we will guide you through all the methodology and finally we discuss the results obtained.

4.1 Experimental Framework

In this section, we will describe the experimental setup used on different scenarios with the same corpora to conduct the experimentation.

First, we describe the task and introduce the software and corpora used in the experimentation; the metric used to assess the results and how the experiments are organized.

Finally, we will compare and discuss the results obtained.

4.2 Task Description

As we mentioned in section 6, there have been several attempts at leveraging monolingual data to improve the quality of machine translation systems in a semi-supervised setting [37]; [38]; & [39]; [40]. Most notably, [1] proposed a very effective data-augmentation scheme, dubbed "back-translation", whereby an auxiliary translation system from the target language to the source language is first trained on the available parallel data, and then used to produce translations from a large monolingual corpus on the target side. The pairs composed of these translations with their corresponding ground truth targets are then used as additional training data for the original translation system.

In this work, we are going to try different configurations to train the translator and different size of the corpora to determine which configuration is the best when we have a large monolingual corpus and a short bilingual corpus.

We will try to do several processes, training different translators using techniques to introduce noise, like inverse translator to see if we can achieve or get closer to the BLEU obtained using the complete parallel corpus.

4.3 Corpora

We tested our proposal with the Europarl[4] [41], which is a collection of *Proceedings* from the *European Parliament,* which are written in all official languages of the European Union and is publicly available on the Internet. This corpus has been used to train all the translators that we will make in this work dividing this in different sizes that will be explained in detail in the following sections of this document.

The main features of the corpus are shown in Table 4.1.

		English	Spanish
Training	Sentences	1.9M	
	Vocabulary	56.8M	61.9M
Development	Sentences	3,003	
	Vocabulary	63,779	62,338
Test	Sentences	3,000	
	Vocabulary	56,089	62,045

Table 4.1: Statistics of the Europarl corpus. In the table are collected the number of sentences and vocabulary size of each partition and language. M stands for millions.

The partition selected as development and test was *news-test2013* that consist in 3K sentences. This are going to be the same during all the experiment for be able to compare the different results.

4.4 Software

We have developed our work using the Open NMT-py toolkit[5]. This toolkit prioritizes efficiency, modularity, and extensibility with the goal of supporting NMT research into model architectures, feature representations, and source modalities, while maintaining competitive performance and reasonable training requirements. The toolkit consists of modeling and translation support, as well as detailed pedagogical documentation about the underlying techniques [42].

OpenNMT was designed with three aims: (a) prioritize fast training and test efficiency, (b) maintain model modularity and readability, (c) support significant research extensibility [42].

[4]http://www.statmt.org/wmt15/translation-task.html

This toolkit also includes a simple reversible tokenizer that (a) includes markers seen by the model that allow simple deterministic detokenization, (b) has extremely simple, language independent tokenization rules. The tokenizer can also perform Byte Pair Encoding (BPE) which has become a popular method for sub-word tokenization in NMT systems [28].

After introducing the software, now we will explain how we set up the hyperparameters to run the experiments.

Firstable, we divided the corpora and try to adjust the hyperparameters with a short corpus (600k instead of 2M) and we made several training until find our parameters and then we used them to train the whole corpora and we obtained the results of the section 4.4.5 with the parameters described below.

4.4.1 Tokenization

Fortunately, as we describe in the section 4.2, OpenNMT[6] has his own toolkit for tokenization that include a simple reversible tokenizer that includes markers seen by the model that allow simple deterministic detokenization and has extremely simple, language independent tokenization rules.

Maybe the most sophisticated tokenizer that we implemented before preprocessing the data was the BPE Codes that we explain in the following section.

4.4.2 BPE Codes

Byte pair encoding or diagram coding is a simple form of data compression in which the most common pair of consecutive bytes of data is replaced with a byte that does not occur within that data. A table of the replacements is required to rebuild the original data. The algorithm was first described publicly y Philip Gage in a February 1994 article "A New Algorithm for Data Compression" in the C Users Journal [29].

One significant advantage of the BPE algorithm is that compression never increases the data size. This guarantee makes BPE suitable for real-time applications where the type of data to be compressed may be unknown. If no compression can be performed, BPE passes the data through unchanged except for the addition of a few header bytes to each block of data. Some algorithms, including LZW, can greatly inflate the size of certain data sets, such as randomized data or pre-compressed files [29].

We first try not to use BPE codes and try the simplest tokenization, but we didn't reach any good result and after applying BPE codes we notice a significance improvement in our result without having to do any modification to our training parameters.

[6] https://github.com/OpenNMT/OpenNMT-py

We have to mention that after using BPE Codes we were able to do the preprocess (see the next section 4.4.3) and then we train and after doing the translation process we needed to do an inverse BPE process to can use the metrics described in section 4.5.

4.4.3 Preprocess

OpenNMT already includes its own preprocess toolkit and we just have to set the source and target sequence length; in our case we use the common sequence length of 70.

Source sequence length	70
Target sequence length	70

Table 4.2: Source and target sequence length used for the preprocess of all the different size of the corpus Europarl before the training process.

4.4.4 Hyperparameters of training process

Using OpenNMT-py has several advantages and we can really set a big amount of hyperparameters to train our models, we didn't explore to much because it would take us a lot time to define the hyperparameters. We just made some variations of the hyperparameters of the table 4.3 and we trained a short corpus just to evaluate which was the best settings, and the best settings are listed below.

Source word vector size	512
Target word vector size	512
RNN size	512
Batch size	50
Optimizer	Adam
Learning rate	0.0002
Learning rate decay	1
Dropout	0
Train steps	1000000
Layers	1
Valid steps	1000
Save checkpoints steps	1000
Label smoothing	0.1
Global attention	mlp

Table 4.3: Settings of the hyperparameters used to train all the models of this work.

4.4.5 Translate

When we finished our train process and have our best model, we translate it using always OpenNMT-py and then we have to do the inverse process of the BPE codes to be able to obtain the results with the standards metrics. In the following table we

shown our best BLEU result using the entire corpora with the settings describe in this chapter. This BLEU result is going to our threshold for complete the task and the main objective of this work.

Corpora with ~ 2M	BLEU	
English-->Spanish	Dev	Test
Europarl	25,19	22,29

Table 4.4: Best BLEU values obtained using the settings described in this chapter with the entire corpora. Dev means for development.

Now, we will try to get closer to the results obtained in table 4.4 by using different combination of corpus size and mixing monolingual and bilingual corpora.

4.5 Metrics

The quality of our initial translation and the difficulty of each task, we used the following well-known metrics.

4.5.1 BLEU

Bilingual Evaluation Understudy (BLEU) computes the geometric average of the modified n-gram precision, multiplied by a brevity factor that penalizes short sentences [43]. BLEU compares the translation generated by a MT system (hypotheses) with a human supervised translation (reference). Formally, it is based on the n-gram concept, it counts the number of candidate words from the hypotheses that appear in the reference and later divides this count by the number of words in the hypotheses.

BLEU is computed as a weighted geometric mean of the different n-gram orders employed. Each n-gram has a weight w_n susch that $\sum_{n=1}^{N} w_n = 1$. Usually weights are set to $w_n = \frac{1}{N}$ and the maximum order is fixed to $N = 4$. These values are taken from [43]. BLEU is computed as the following equation:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n log p_n\right) \tag{4.1}$$

4.5.2 TER

Translation Error Rate (TER): Computes the number of words edit operations (insertion, substitution, deletion and swapping), normalized by the number of words in the final translation [44].

TER is computed as follows:

$$TER = \frac{\#of\ edits}{\#of\ reference\ words} \tag{4.2}$$

All edits have equal cost. To compute number of edits a dynamic programming algorithm is used.

4.5.3 BEER

BEER is a sentence level metric that can incorporate a large number of features combined in a linear model. Novel contributions are (1) efficient tunning of large number of features for maximizing correlation with human system ranking, and (2) novel features that give smoother sentences level scores [15].

In the work of [15] from the metrics that participated in all language pairs on the sentences level on average BEER has the best correlation with the human judgment.

4.6 Experimental Set-Up

In this section, we are going to describe all the flow process of the experimentation.

Using the corpora described in section 4.3, we divided it in two equal parts, one for the bilingual experimentation and the other for the monolingual experimentation as shown in the following figure.

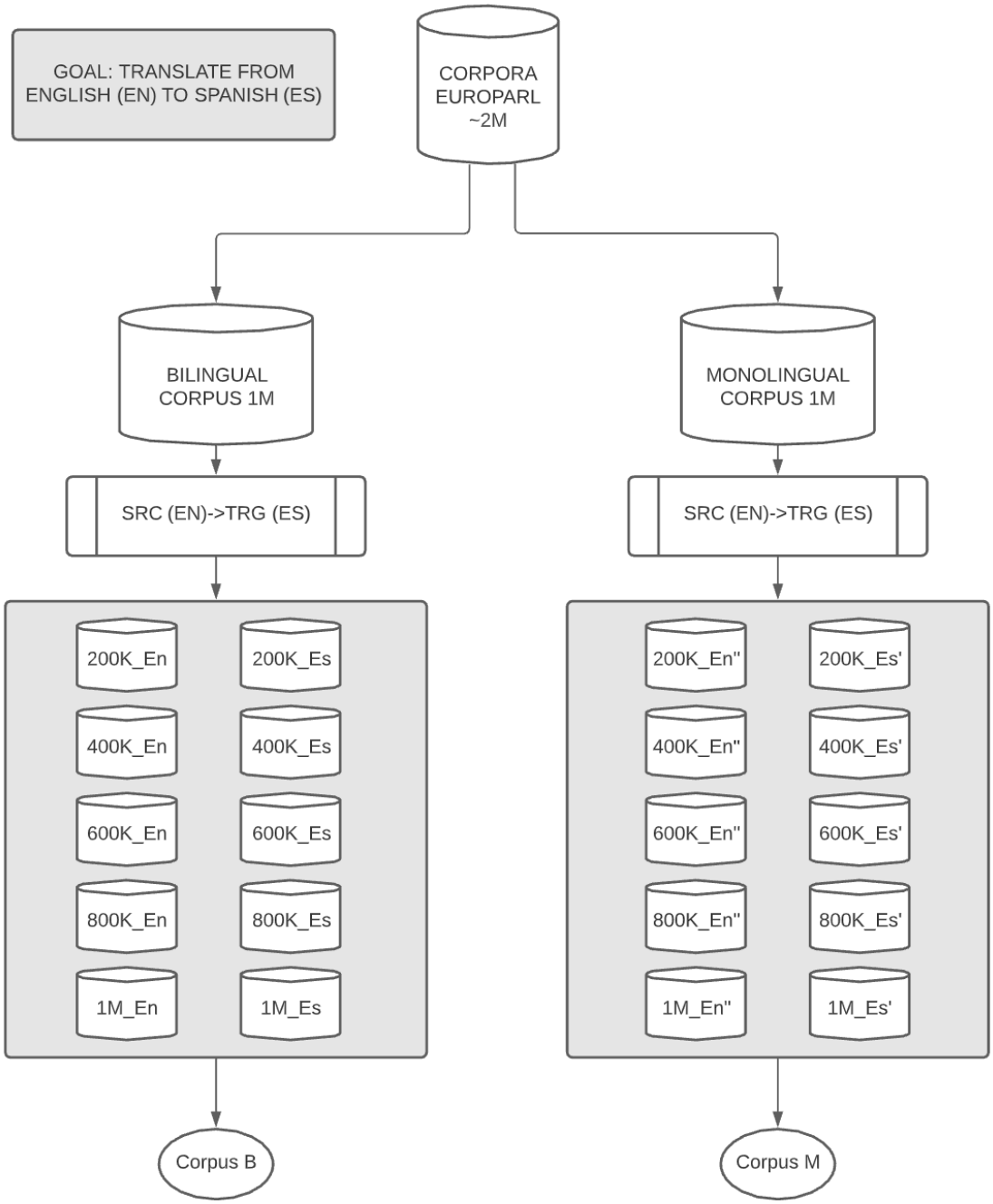

Figure 4.1: Main division of the corpora for the whole experiment reminding the goal that is to translate from English (En) to Spanish (Es). Pay special attention with **the first apostrophe and double apostrophe** in each subdivision of the corpus to indicate the source and target side.

Before continue with the experiment we first have to find and acceptable hyperparameters to train our translator using the software OpenNMT-py (see section 4.4) because this are going to remind equal for the rest of the experiment even if we don't apply to much sophisticated techniques for preprocess the data and train but this is not the main goal of the work and trying to fit the BLEU score to the ones obtained in other works like in [10], [45], [24] and [1] it would have stopped as for a long time.

Having found the configuration set for preprocess and train the data we moved to train the translator number 1 with the corpus B as follow:

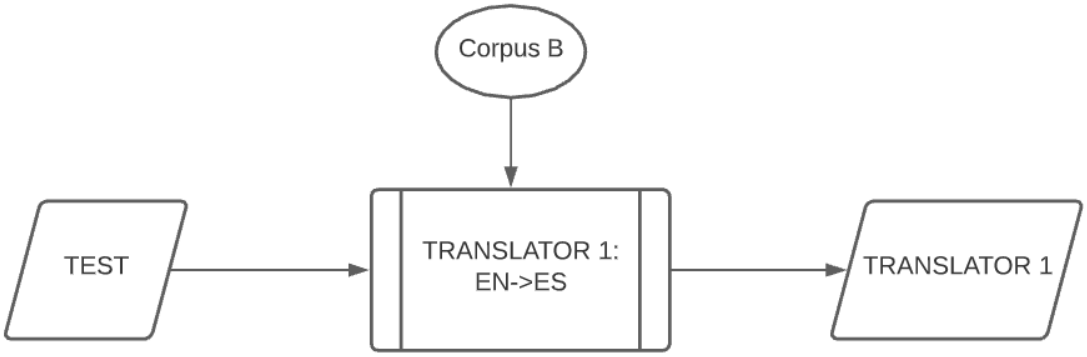

Figure 4.2: First direct translator (from English (EN) to Spanish (ES)) trained with the "Corpus B".

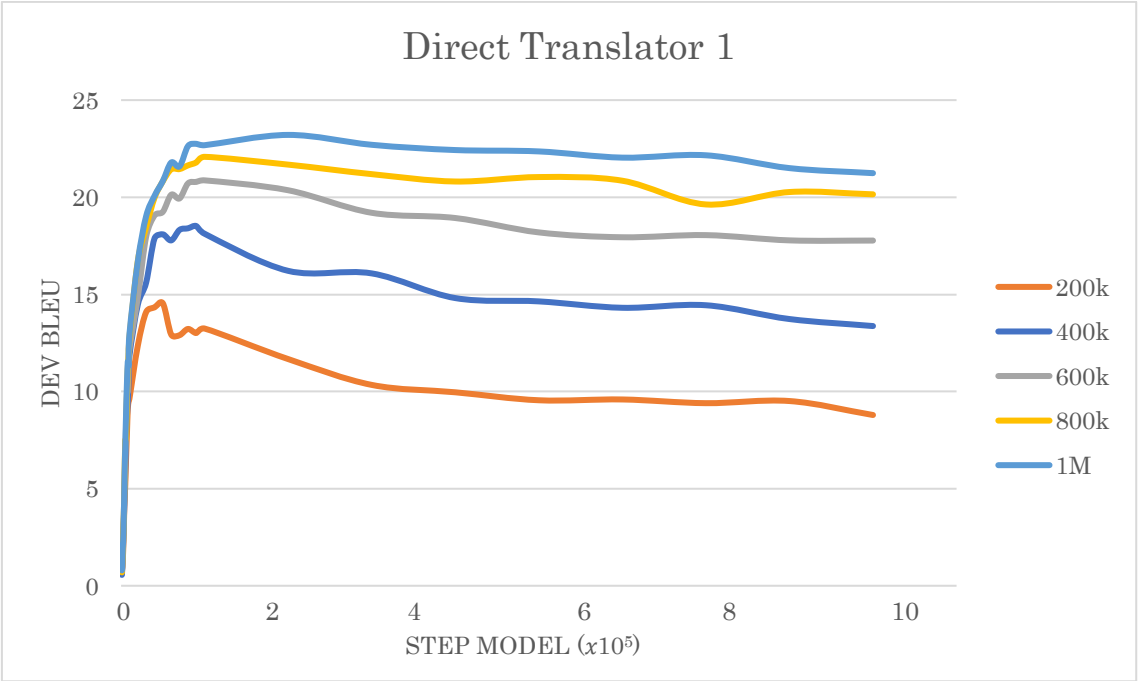

Figure 4.3: First direct translator (from English (EN) to Spanish (ES)) trained with the "Corpus B" and here we show the BLEU obtained with the development for the different size of corpus (from 200k to 1M).

After doing this process, we can notice with what step model we obtained the best BLEU results, taking notes of this values we proceed to obtained results using the metrics of TER and BEER and we got the results from the table 4.5.

TRANSLATOR 1

Training		Step Model	Development	
Source English	Target Spanish		TER	BEER
200K	200K	50000	72.60	50.83
400K	400K	90000	66.80	51.59
600K	600K	100000	62.70	53.39
800K	800K	100000	61.70	54.41
1M	1M	200000	60.50	54.61

Table 4.5: This table represent the different size of the corpus that we used for the direct translation number one and the best step model which we got the best BLEU result. With this step models we algo got the metrics of TER and BEER. K stands for thousands and M for millions.

We did this last step with every process of the experiment and then we summarize the results in only one table as we can see in table 4.6.

TRANSLATOR 1

Training		Development		
Source English	Target Spanish	BLEU	TER	BEER
200K	200K	14.54	72.60	50.83
400K	400K	18.52	66.80	51.59
600K	600K	20.87	62.70	53.39
800K	800K	22.08	61.70	54.41
1M	1M	23.21	60.50	54.61

Table 4.6: This table contains the best results for the development for the BLEU, TER and BEER metric for the experiment with the translator number one. K stands for thousands and M for millions.

Then we try to introduce some noise training an inverse translator (Es→En) using the "Corpus B" and with this translator we will obtain a pseudo English corpus and then we will use this results together with the "Corpus M" using the target part (Spanish side) to form a pseudo bilingual corpus (P).

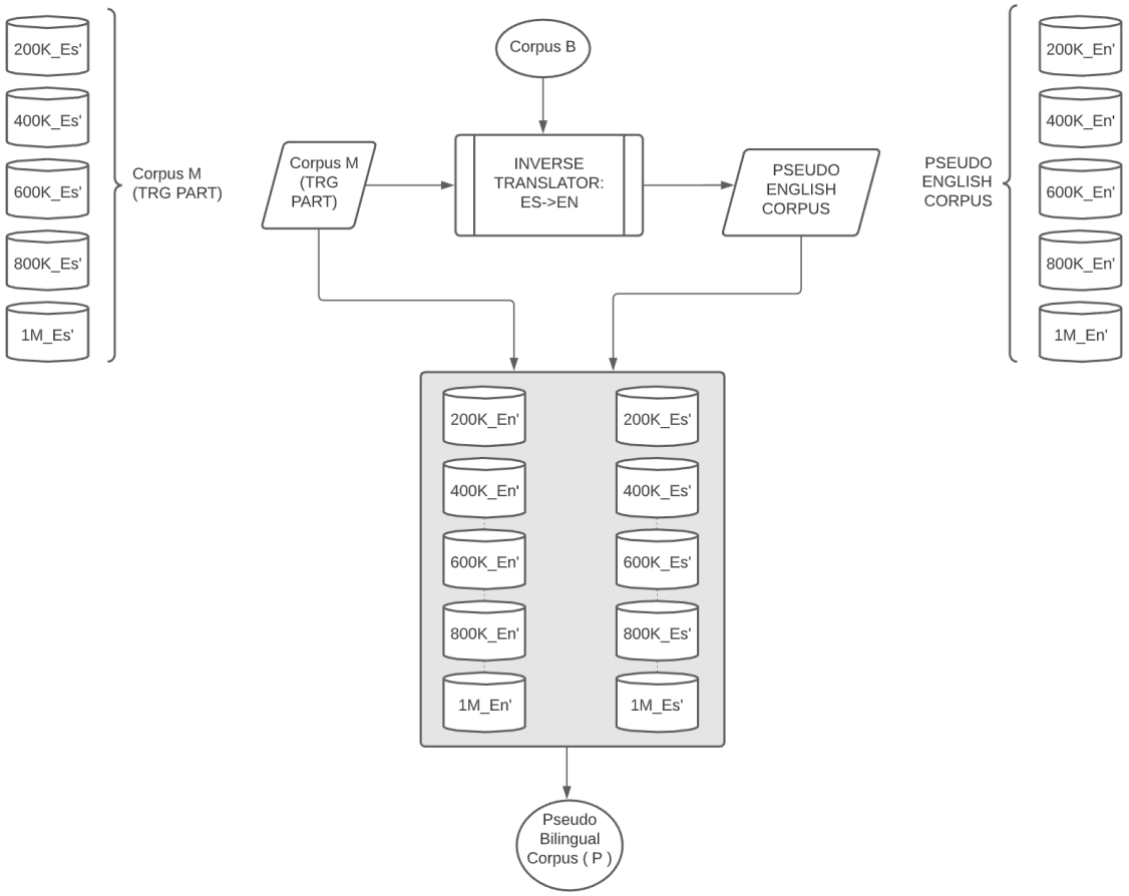

Figure 4.4: This figure explained how we formed the "Pseudo Bilingual Corpus (P)" using and inverse translator as a noise technic, training it with "Corpus B" and having the target part of the "Corpus M" as input. "En" refers to English and "Es" to Spanish. K stands for thousands and M for millions.

While doing this task, we thought that it would be interesting to obtained different results with different combination of the best training model to translate the different inputs. Explaining more this idea, we will going to train our inverse translator using the Corpus B and we will going to keep the best training model for each size of the corpus B and then used the best models to do different combination of translation and find if it is possible to get better translation using the model obtained with the corpus size of 200k when the input size is 400k (giving just an example of all the possible combination).

In the following figure we explained all the combinations made in this phase of the experimentation.

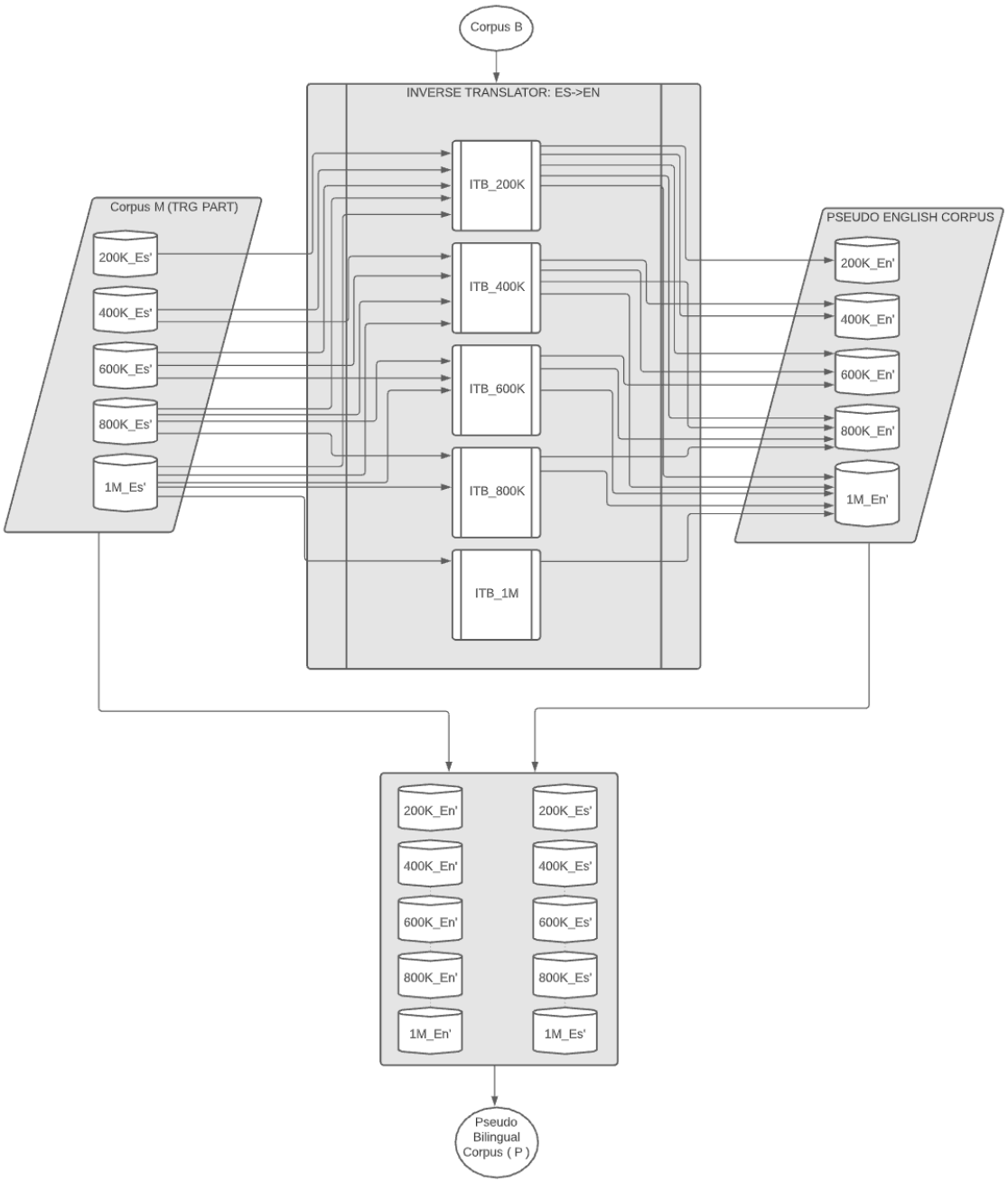

Figure 4.5: This is a detailed explanation of the figure 4.4 showing all the combination that we made using the best training model of the inverse translation for the 200k, 400k, 600k, 800k and 1M. We identify each model by "ITB_#corpus_size" where "ITB" means "Bilingual Inverse Translation" and for the names of the new "Pseudo Bilingual Corpus (P)" "En" refers to English and "Es" to Spanish. K stands for thousands and M for millions.

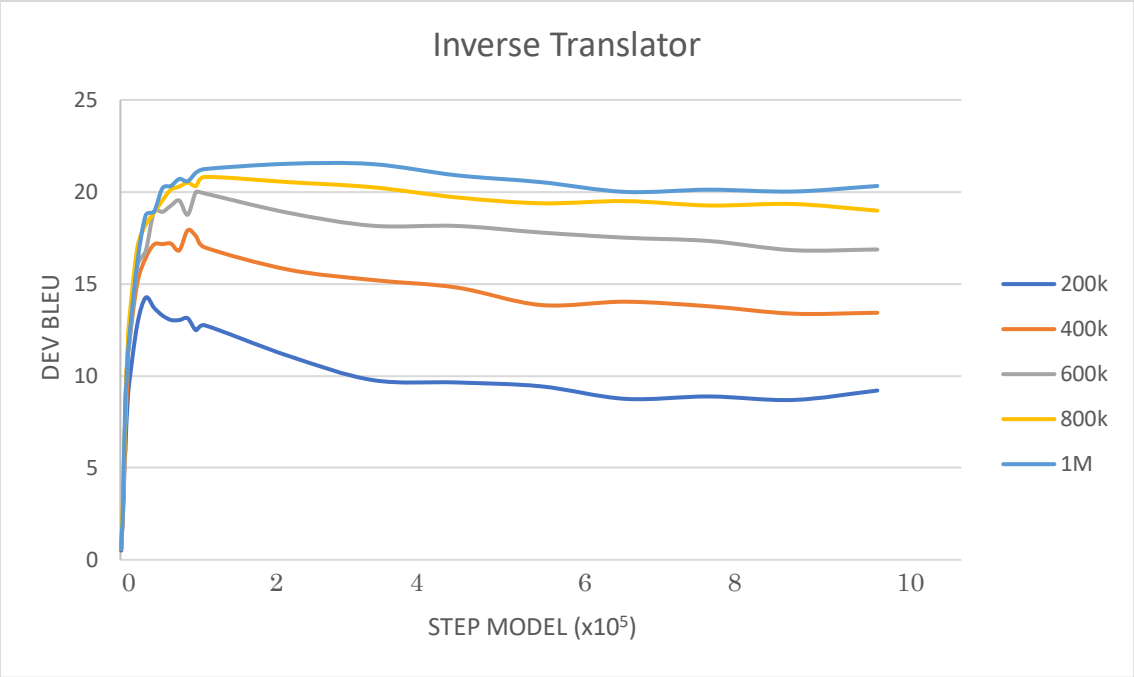

Figure 4.6: This figure represents the different BLEU result that we got while training the inverse translator with the different size of the corpus, and knowing this, is how we choose the best step model for each size of the corpus. These selections are the ones that we use in the figure 4.5 with the name ““ITB_#corpus_size”. K stands for thousands and M for millions.

After comparing the results, we decided to save just the best value and compare with which size of training model we obtained that result, just to noticed if we could find a better result doing those combinations.

We did the mentioned above by constructing a new translator number four that will be trained with the new pseudo bilingual corpus (P) as stated in the next figure.

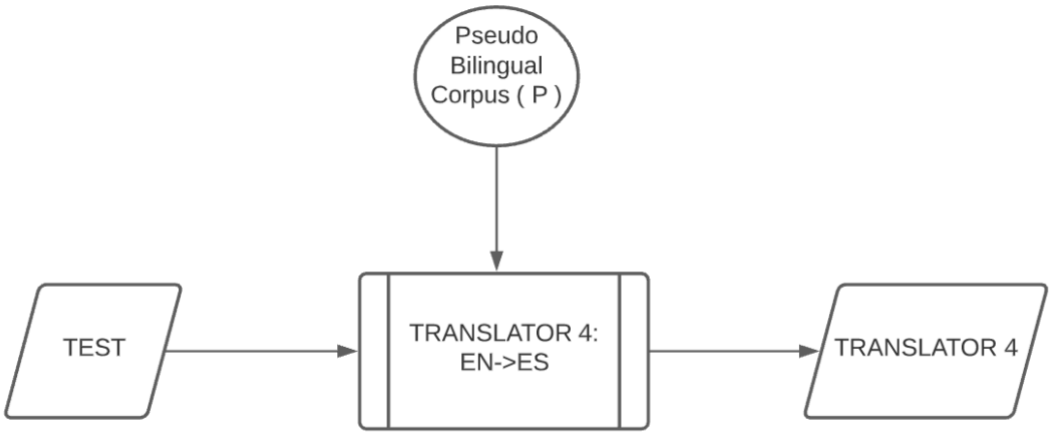

Figure 4.7: Translator number four. Source: English and Target: Spanish using the “Pseudo Bilingual Corpus (P)” for the training process.

As we described in the past sections, we wanted to make different combinations to evaluate if we could improve any result.

In the following figures we compare each combination with the distinct models of the inverse translator described in the figure 4.4 and 4.5.

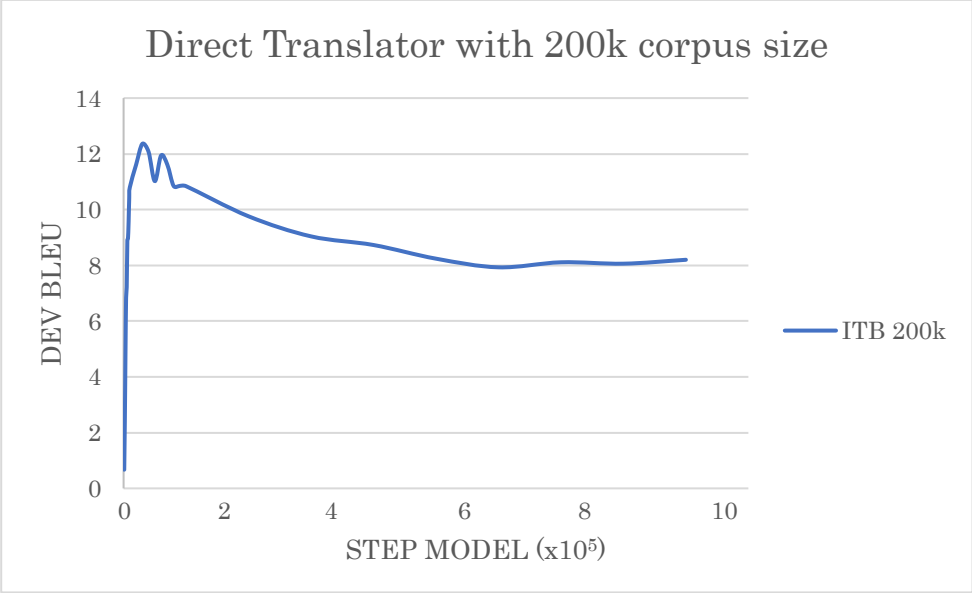

Figure 4.8: Development BLEU results obtained with different step models using the "ITB 200k" model in the training process to translate the 200k corpus size. "ITB" means "Bilingual Inverse Translation"

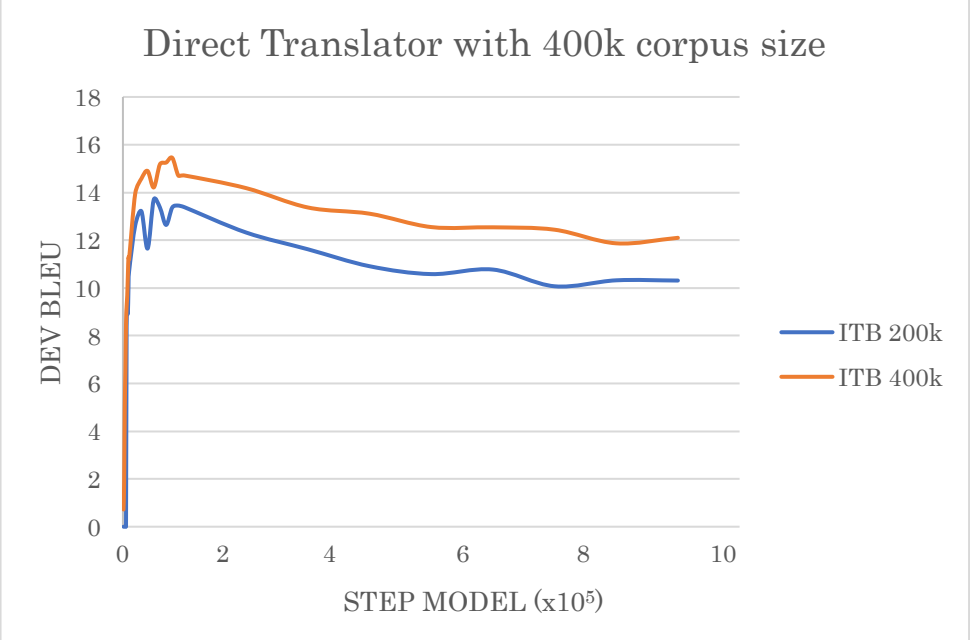

Figure 4.9: Development BLEU results obtained with different step models using the "ITB 200k and ITB 400k" models in the training process to translate the 400k corpus size. "ITB" means "Bilingual Inverse Translation"

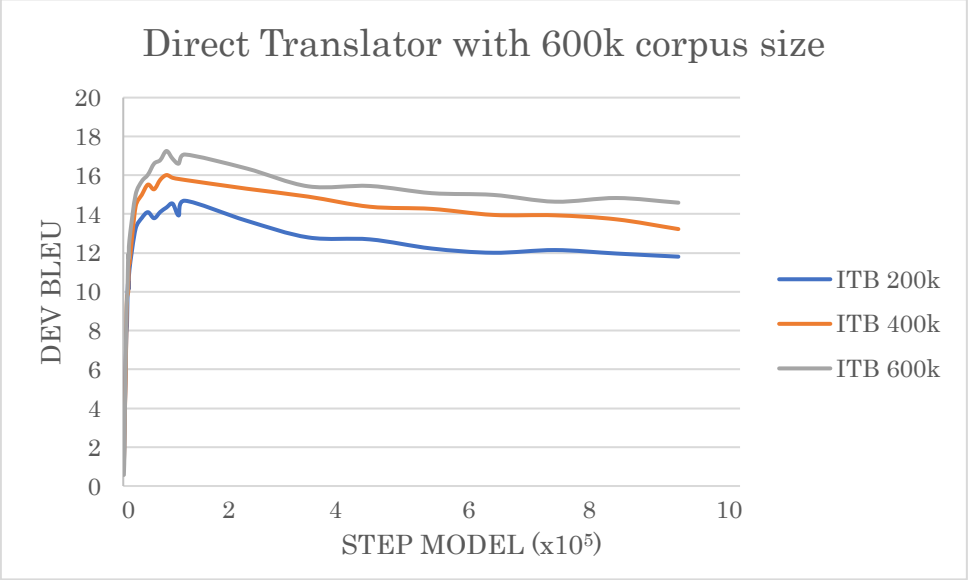

Figure 4.10: Development BLEU results obtained with different step models using the "ITB 200k, ITB 400k, ITB 600k" models in the training process to translate the 600k corpus size. "ITB" means "Bilingual Inverse Translation"

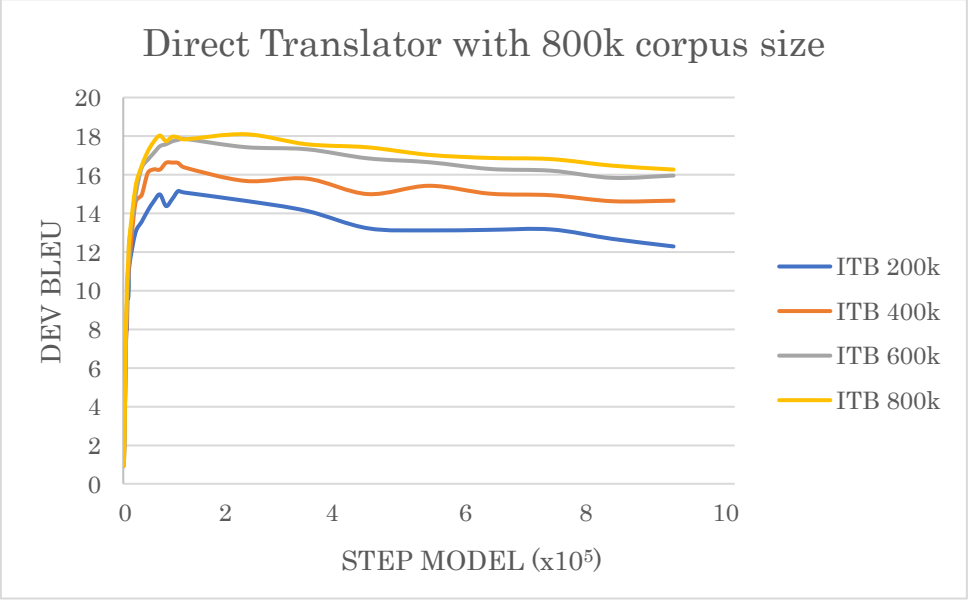

Figure 4.11: Development BLEU results obtained with different step models using the "ITB 200k, ITB 400k, ITB 600k and ITB 800k" models in the training process to translate the 800k corpus size. "ITB" means "Bilingual Inverse Translation"

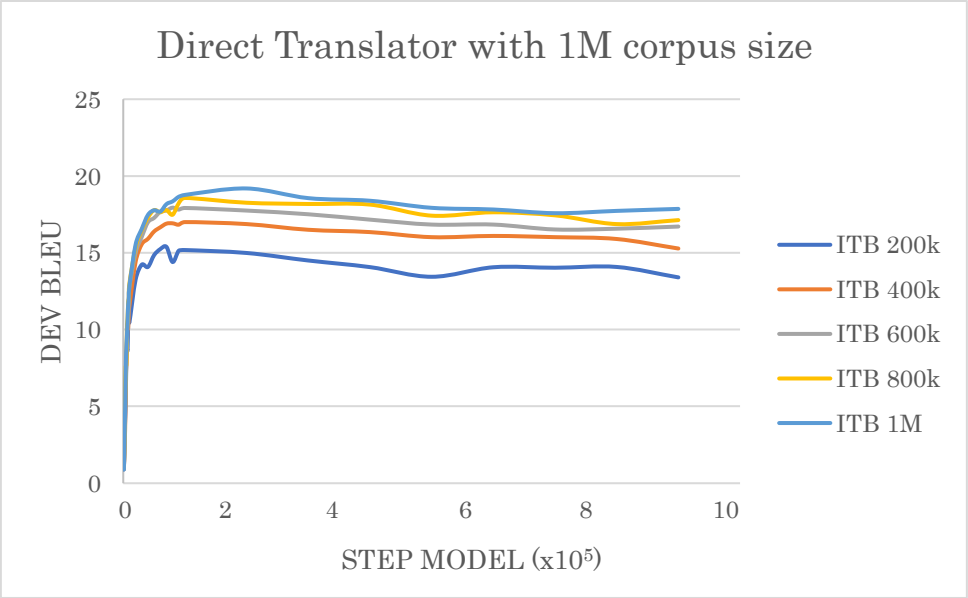

Figure 4.12: Development BLEU results obtained with different step models using the "ITB 200k, ITB 400k, ITB 600k, ITB 800k and ITB 1M" models in the training process to translate the 1M corpus size. "ITB" means "Bilingual Inverse Translation"

The process that we have just mentioned, we called it "translator for control" before following the next steps.

We didn't know if this process was going to present good or bad results. Fortunately, we were able to move to the following step.

Now, this becomes more interesting. We plan to mix the "Corpus B" with the "Pseudo Bilingual Corpus (P)" to construct the translator number two. We shuffled the pairs of corpora and made a script to adjust the both pair of corpora (English and Spanish) for be a parallel corpus after joining and doing the shuffle, see the figure 4.13 for the name references to identify the new pairs of corpora for the experiment.

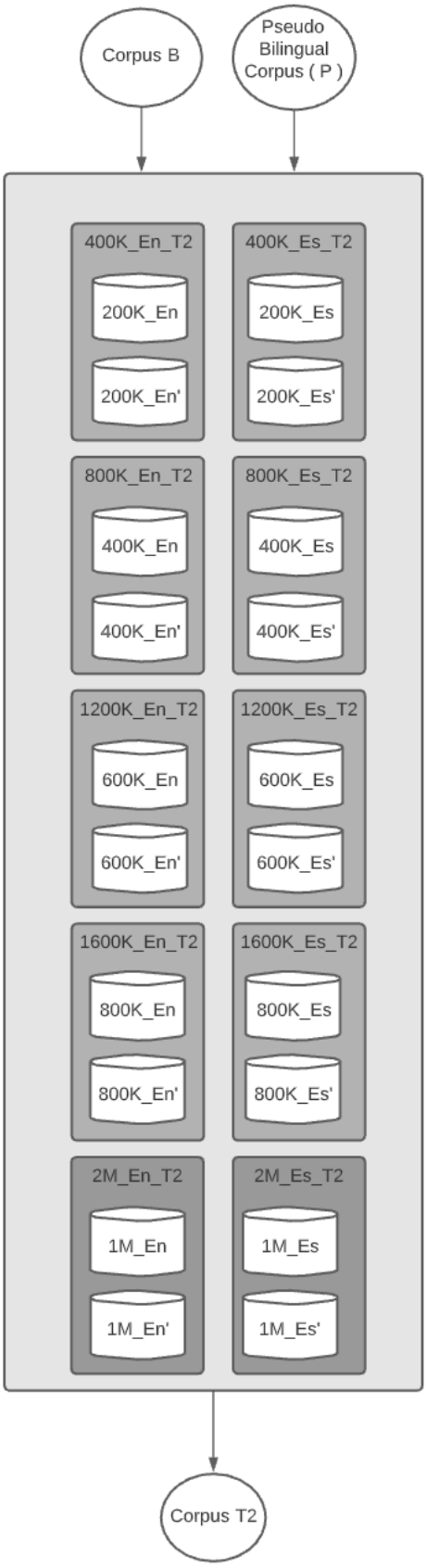

Figure 4.13: Here we show how we mixed the "Corpus B" and the "Pseudo Bilingual Corpus (P)" to form the new "Corpus T2", in this figure you have to pay special attention on the first and double apostrophe for be able to identify the way we merge each part and after that we shuffle and adjust the pair of the new corpus. We named

this new pairs as "#Size_#Lenguage_T2" to identify the new join and shuffle of pairs. K stands for thousands and M for millions.

Using the new "Corpus T2" we trained the translator number two and we got result for the development and test.

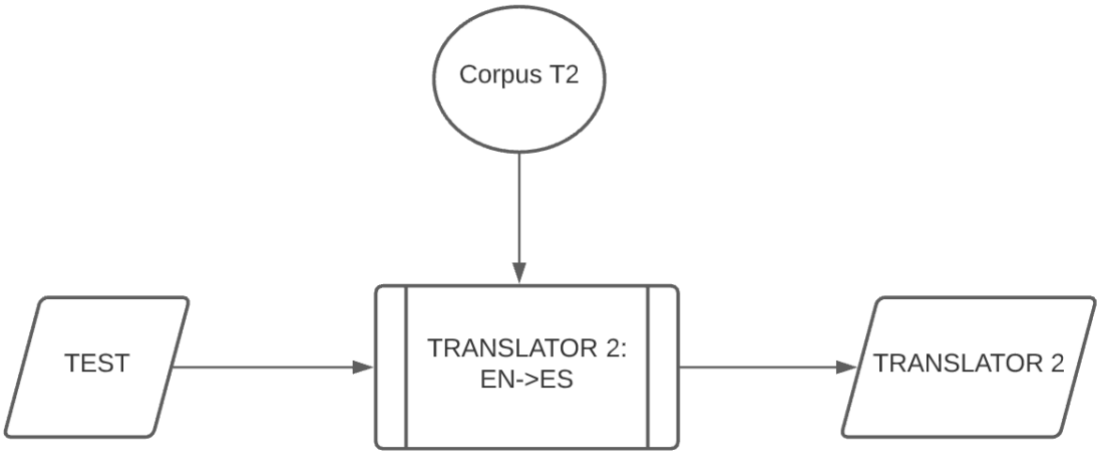

Figure 4.14: Translator number two trained with the new "Corpus T2"

TRANSLATOR 2

Training		Development		
Source English	Target Spanish	BLEU	TER	BEER
400K	400K	15.55	71.90	47.79
800K	800K	19.22	66.70	50.42
1200K	1200K	20.88	63.40	52.03
1600K	1600K	21.00	63.90	52.00
2M	2M	23.39	59.10	54.00

Table 4.7: This table contains the best results for the development for the BLEU, TER and BEER metric for the experiment with the translator number two. K stands for thousands and M for millions.

Finally, we shuffle the pair of the corpus "Corpus B" with "Corpus M" following the same criteria using in the figure 4.13 and using the same script for shuffling the corpora as we did to build the "Corpus T2".

We create now the "Corpus T3" (see the figure 4.15) that we used to train the translator number three.

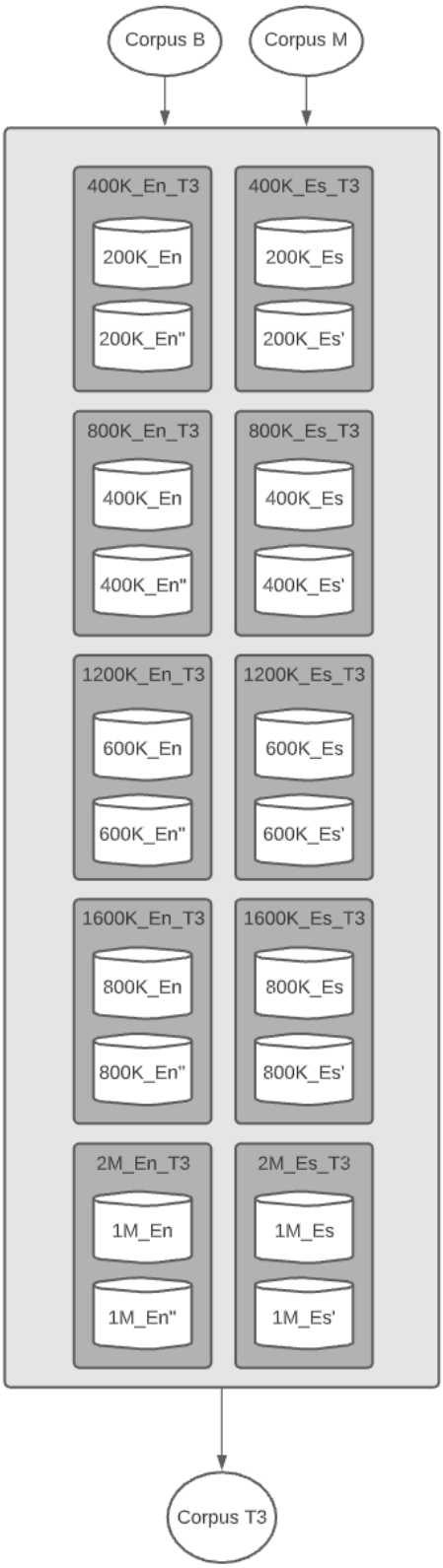

Figure 4.15: Here we show how we mixed the "Corpus B" and the "Corpus M" to form the new "Corpus T3", in this figure you have to pay special attention on the first and double apostrophe for be able to identify the way we merge each part and after that we shuffle and adjust the pair of the new corpus. We named this new pairs as "#Size_#Lenguage_T3" to identify the new join and shuffle of pairs (as we did for the "Corpus T2"). K stands for thousands and M for millions.

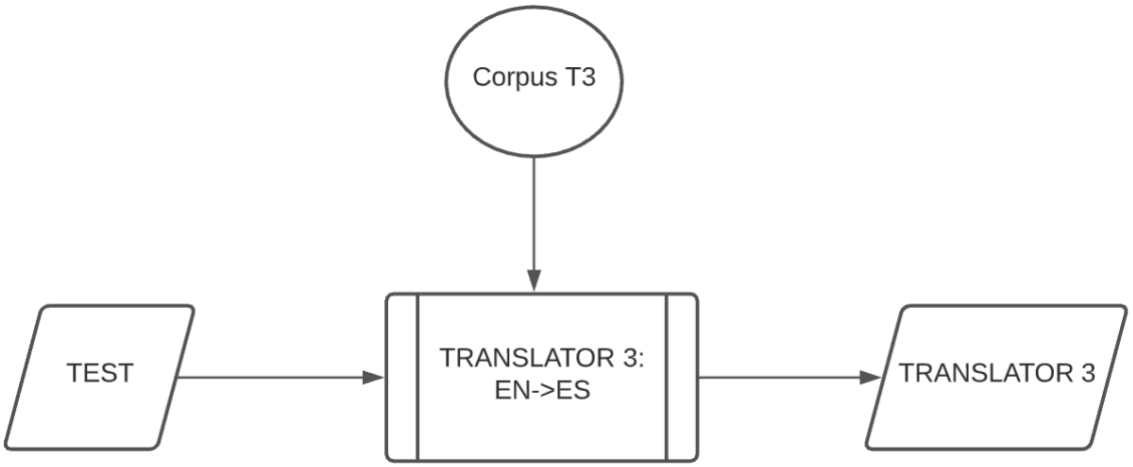

Figure 4.16: Translator number three trained with the new "Corpus T3".

TRANSLATOR 3

Training		Development		
Source English	**Target Spanish**	**BLEU**	**TER**	**BEER**
400K	400K	19.00	66.80	50.91
800K	800K	21.16	63.40	51.99
1200K	1200K	22.47	60.60	53.33
1600K	1600K	23.53	60.50	53.96
2M	2M	24.06	60.10	53.64

Table 4.8: This table contains the best results for the development for the BLEU, TER and BEER metric for the experiment with the translator number three. K stands for thousands and M for millions.

4.7 Results

In this section of the chapter we will discuss our results obtained with the experimental set-up of the past section.

As we can see in the figure 4.3, we got different results of the BLEU using the development corpus and as expected, when we used the half of the entire corpora we got better results than using just a little part (like 200k).

Using this half of the bilingual corpora and doing our first translator (English to Spanish) we acquire results for the well know metrics of BLEU, TER and the newest

BEER. In table 4.6 you can see the results for the development data and in the following table we present summarize the results using the test data.

TRANSLATOR 1

Training		Test		
Source English	Target Spanish	BLEU	TER	BEER
200K	200K	13.38	74.60	47.53
400K	400K	16.78	69.00	50.83
600K	600K	18.75	65.90	52.31
800K	800K	20.10	64.70	53.55
1M	1M	20.64	63.20	53.61

Table 4.9: This table contains the best results for the test data for the BLEU, TER and BEER metric for the experiment with the translator number one. K stands for thousands and M for millions.

The metrics are different each other as we mentioned in section 4.5 but putting all these together, we can measure our translation quality.

Remember the value that we got in table 4.4 using the entire corpora, the baseline is 22,29 of BLEU, and in this translator using just one half of the corpora we have a BLEU of 20,64 that is not so far (a difference of 1,65 points). But when we just use 200k we got a BLEU of 13,38 that is evidently a bad result (8,91 points of difference). We keep all this that because this is just the beginning of the whole experiment and then we will compere if applying different techniques, we are able to improve these results.

In the column of the TER results in table 4.9 (where lower is better) our best result is still with 1M of sentence (half of corpus) and if we see the column of the BEER results (higher is better) in the same table, we agree that with much more data of the corpus we can improved the quality of our translator.

We cannot find a perfect correlation between BLEU, TER and BEER, even if they all measure our translation quality, we can just validate our results in every experiment with different corpus size.

After doing the first direct translator, we wanted to introduce a kind of noise by doing an inverse translator in the middle of the process.

The task of the inverse translator is better described in the 4.6 section by reading the figure 4.5.

This may be one of the most challenging translators that we made, and this part of the process took a lot of machine time for all the training corpus that we made for save only the best training model and then used them for several translation to obtain the new “Pseudo English Corpus” and build a new “Pseudo Bilingual Corpus (P)” to train a new direct translator (number four) with this corpus.

By analyzing the figure 4.5, you can see that inside of the box of "inverse translator" there are another 5 boxes and each one of them represent the best training model that we got with the different size of the "Corpus B" and then we made all the possible combination (these are marked in the figured 4.4) to translate the target part of the "Corpus M" (see figure 4.1) and the result obtained we called it "Pseudo English Corpus" and with the input and output of the inverse translator we assemble the "Pseudo Bilingual Corpus (P)"

In the figure 4.6 is feasible to see the work process that we did to select the best training model that you can see with the name "ITB_#corpus_size" in the diagram of the figure 4.5.

Subsequently, using the new corpora (Pseudo Bilingual Corpus (P)) we train our direct translator number four. (see figure 4.5) and we pass through all the training and translator process as you can see the results obtained with the development data on figures 4.8 to 4.12. We selected the best step model to measure our translator. Nevertheless, on the figures it's viable to compare every combination and it's evident that the best results are when the corpus size that we are trying to translate match with Its training model (for example: when translating the corpus size of 400k, we got better BLEU result with the train model ITB 400K than ITB 200K, see fig. 4.9)

Table 4.10 shows the different results achieved with the multiple combinations made with the training models. You can see that is marked the best result that does match with the corpus size and its training model size.

DIRECT TRANSLATOR 4

Corpus Size	Train Model	Step Model	BLEU TEST
200K	ITB 200K	30000	11.32
400K	ITB 200K	50000	12.06
	ITB 400K	80000	13.56
600K	ITB 200K	100000	13.04
	ITB 400K	70000	14.47
	ITB 600K	70000	15.18
800K	ITB 200K	90000	13.4
	ITB 400K	80000	15.11
	ITB 600K	100000	15.43
	ITB 800K	200000	16.27
1M	ITB 200K	70000	13.59
	ITB 400K	100000	14.89
	ITB 600K	80000	15.77
	ITB 800K	100000	16.16
	ITB 1M	200000	16.99

Table 4.10: This table contains the best results for the test for the BLEU metric for the experiment with the inverse translator number four. "ITB" means "Bilingual Inverse Translation"; K stands for thousands and M for millions.

Afterward this laborious procedure of the "translator for control" let us following with the next step but now using just one training model (the best) and by knowing

the results of table 4.10 we determine that will not be necessary to do all those combination for the next translations and we can save machine time.

Look back on figure 4.5 we build the "Pseudo Bilingual Corpus (P)" and with the "Corpus B" (see the subdivision of the corpus of figure 4.1) we build the new "Corpus T2" as described in figure 4.13. This last figure is important to keep in mind to understand the new labels of the corpus that we mention on the results of table 4.7 and in table 4.11.

Fortunately, with the new translator two we improved our translation comparing this result with the ones we got in with the translator one (table 4.9) recalling that these results are with a pseudo bilingual corpus injecting some noise by using the inverse translator. In table 4.11 we can know that mixing half of the entire corpora and half of pseudo bilingual corpus (P) we achieve better results instead of mixing other proportions even if those results were also improved.

TRANSLATOR 2

Training		Test		
Source English	Target Spanish	BLEU	TER	BEER
400K	400K	14.12	73.40	47.84
800K	800K	17.47	69.00	50.37
1200K	1200K	18.47	66.00	51.20
1600K	1600K	19.21	65.50	51.57
2M	2M	21.08	61.40	53.71

Table 4.11: This table contains the best results for the test for the BLEU, TER and BEER metric for the experiment with the translator number two. K stands for thousands and M for millions.

To resolve the main objective of this work, now we join and shuffle the "Corpus B" and "Corpus M" (see the subdivision of the corpus of figure 4.1) as we shown in figure 4.14 and we get closer with our threshold using the development set of data mixing half of "Corpus B" and half of the "Corpus M" obtained a BLEU of 24,06 comparing with the 25,19.

Table 4.12 resumes the calculations of the translator number three using the test data, mercifully we improved the results comparing these with the ones we got with the translator one and two.

We get a little closer to the threshold of the BLEU of 22,29 obtained with the full bilingual corpora, since we used half of bilingual corpora and half of monolingual corpora, we can achieve a BLEU of 21,18. Moreover, contrasting the rest of the calculations with different sizes, all of them were also upgraded.

TRANSLATOR 3

Training		Test		
Source English	Target Spanish	BLEU	TER	BEER
400K	400K	17.25	66.90	50.43
800K	800K	19.08	65.50	51.60
1200K	1200K	20.41	62.90	53.16
1600K	1600K	20.68	63.10	53.23
2M	2M	21.18	63.10	53.23

Table 4.12: This table contains the best results for the test for the BLEU, TER and BEER metric for the experiment with the translator number three. K stands for thousands and M for millions.

In the following figures (from 4.17 to 4.19) we did a comparison between the results obtained with translator 2 and translator 3 for the three metrics used in these experiments. You have to remember the diagram of the figure 3.4 when we explained in a general way the methodology implemented. As you can see, in translator two we used back translator as we explained above and in translator three, we just mixed different ratios of bilingual and monolingual data.

Here the point is very interesting because we generally obtained better results in translator number three (without back translator).

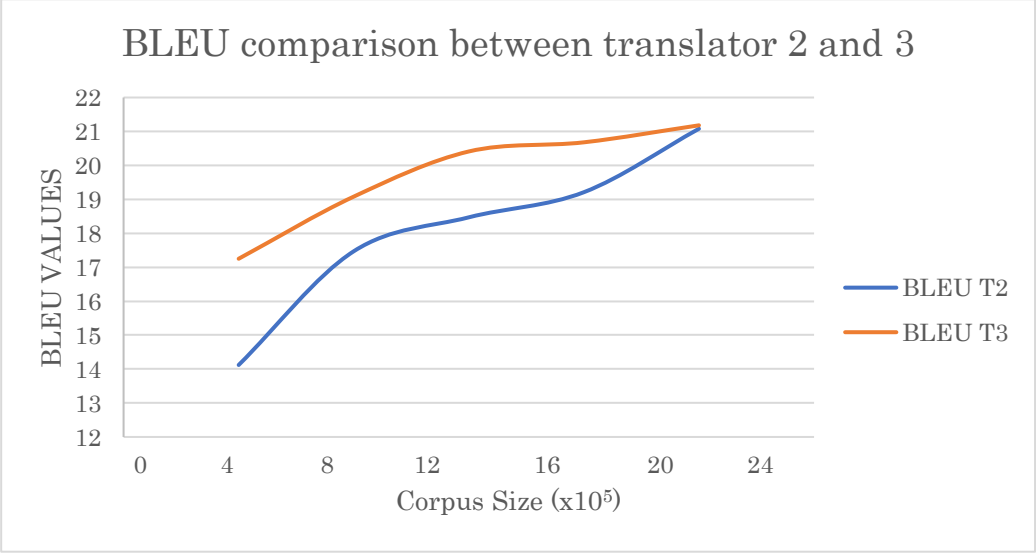

Figure 4.17: Comparison of the metric BLEU between translator 2 (T2 in blue color) and translator 3 (T3 in orange color) (higher is better).

By seeing the figure 4.17, we can notice that generally we obtained better results with translator three than translator two, but finally both are very close with the best result achieve with the whole corpus, the difference is just 0.10 points.

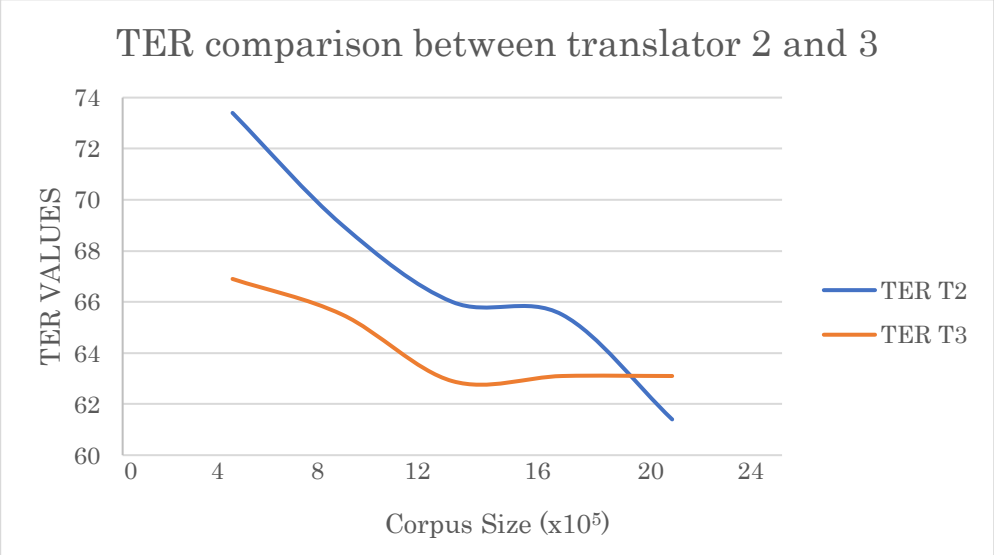

Figure 4.18: Comparison of the metric TER between translator 2 (T2 in blue color) and translator 3 (T3 in orange color) (lower is better).

In the figure above (4.18) we noticed as in figure 4.17 that generally, we obtained better results of TER values in translator three, but at the end with the whole corpora, translator two got better results. It means that by using back translation and the TER metric, it seems to improve the translation quality. The differences between translator two and three is 1.7 points.

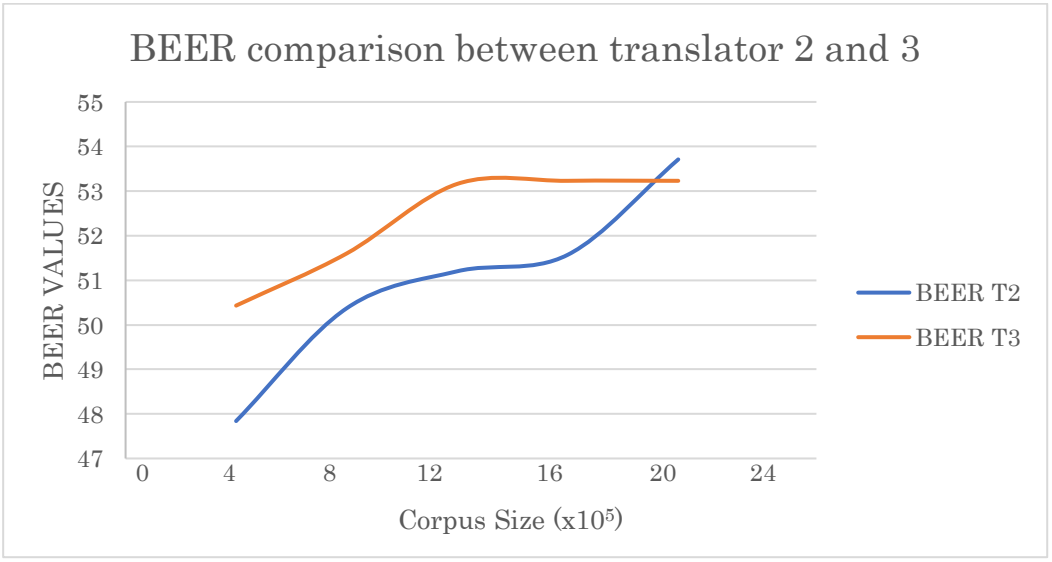

Figure 4.19: Comparison of the metric TER between translator 2 (T2 in blue color) and translator 3 (T3 in orange color) (higher is better).

Comparing the results between translator two and three using the newest metric, BEER (see figure 4.19), we see the same behavior as in figure 4.17 and 4.18. Translator three generally shows better results with the different corpus size, but by the end with the entire corpora, we got better results with translator two.

By doing the previous analysis, we conclude that we achieve better results when we use back translator to build our pseudo bilingual corpus, these results correspond to the translator number two (see table 4.11 for the results) see the figure 3.4 to get the big picture of the experiment and 4.5 and 4.13 for more details.

The main objective of this work was to build a translator by training it with monolingual and bilingual corpora, not just bilingual as we are used to it and find in any paper of machine translation. Nowadays, we don't find too much work done with this kind of corpora even if we have access of plenty free monolingual corpora.

Driving throw all these processes we found that it's possible to build an acceptable translator by using just half of the corpus bilingual and the other half monolingual, but if we are in a situation that we don't have half and half of the corpus, as we demonstrate in table 4.11 and 4.12, we can still make a translator with other proportions and achieve reasonable results.

The next steps that we would like to try, were to replicate this work by using only our best results to build a translator using other pair of languages and try to explore different corpus size combinations. This will be a future work because the actual master thesis took too much machine time for all the training procedures and we were doing this considering that it could be a complete failure (not getting closer to the threshold BLEU value) but now that we realize that we still have a complete world to explore by using just monolingual corpora.

CHAPTER 6
Conclusions

5.1 Conclusions

5.1 Conclusions

In this master thesis, we proposed to divide a full bilingual corpus in two equal parts, the first side we used it as a bilingual corpus and the other one we used it just as a monolingual corpus. Putting forward to build a translator using monolingual corpus but also a part of bilingual corpus to translate from English to Spanish we did the procedure presented in section 4.6, obtaining very encouraging results by trying to build several translations with different proportions of the corpora and using the technique of back translator. These results show that, our methodology gets closer to the translation quality that use only bilingual corpora.

In this experiment, the results obtained by our methodology were just different from one point of BLEU comparing with the baseline, in our worst case of the proportions of the data, the difference was of five points of BLEU.

Overall, results show that we are in the right path to develop a methodology that improves the translation quality by using monolingual corpus and not only bilingual corpus, but we still have some more work to do.

5.2 Future Work

Among the future work, it would be interesting to experiment with more diverse corpora and try to use different criteria for mixing the corpora and take advantage of the free monolingual corpora that all of us could have access on internet.

Another thing to do is to incorporate a new inverse translator into our methodology to have more than one that plays the role of injecting noise to train our translators. This way, we could build a better translator.

Finally, in this thesis we have used a 2K sentence corpora, it could be interesting to try this methodology by using a bigger corpus. Having the advantage of the methodology used, a further engaging project could be to do a more sophisticated pre-process of the data and improve the initial hyperparameters for training and then follow the same experimental set up of this master thesis.

CHAPTER 6
Bibliography

6. Bibliography

[1] R. Senrich, B. Haddow and A. Birch, "Improving neural translation models with monolingual data.," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,* pp. 86-96, 2016.

[2] M. Tallerman and K. R. Gibson, "The Oxford handbook of language evolution.," Oxford, New York: Oxford University Press., 2012.

[3] M. Müller, M.A, "The theoretical stage, and the origin of language. Lecture 9," in *Lectures on The Science of Language*, New York: Charles Scribner, 124 Grand Street., Royal Insitution of Great Britain, 1961, pp. 287-328.

[4] M. H. Christiansen and S. Kirby, "Language Evolution: The Hardes Problem in Science?," in *Language Evolution*, Oxford, New York: Oxford University Press., 2003, pp. 77-93.

[5] Y. Wang, C. Zhai and H. Hassan Awadalla, "Multi-task Learning for Multilingual Neural Machine Translation," *arXiv:2010.02523v1,* 2020.

[6] P. Koehn and R. Kowles, "Six challenges for neural machine translation.," *Proceedings of the First Workshop on Neural Machine Translation. ,* pp. 28-39, 2017.

[7] F. Burlot and F. Yvon, "Using Monolingual Data in Neural Machine Translation: a Systematic Study," *Proceedings of the Third Conference on Machine Translation,* vol. 1, pp. 144-155, 2018.

[8] C. D. Vu Hoang, P. Koehn, G. Haffari and T. Cohn, "Iterative Back-Translation for Neural Machine Trasnlation," *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation.,* pp. 18-24, 2018.

[9] D. Ortiz-Martínez, "Advances in Fully-Automatic and Interactive Phrase-Based Statistical Machine Translation," PhD thesis, Universidad Politécnica de Valencia, 2001.

[10] M. Domingo, "Interactive Post-Editing in Machine Translation.," Master's Thesis. Universidad Politécnica de Valencia, 2015.

[11] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra and R. L. Mercer, "The mathematics of machine translation: Parameter estimation.," *Computational Linguistics,* vol. 2, no. 19, pp. 263-311, 1993.

[12] F. Josef Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation.," *Proceedings of the 40th Annual Meeting,* pp. 295-302, 2002.

[13] P. Koehn, F. Josef Och and D. Marcu, "Statistical phrase-based translation," *Proceedings of the Human Language Technology,* vol. 1, pp. 48-54, 2003.

[14] R. Zens, F. Josef Och and H. Ney, "Phrase-based statistical machine translation.," *Advances in artificial intelligence. 25 Annual German Conference,* vol. 2479 of LNCS, pp. 118-32, 2002.

[15] M. Stanojević and K. Sima'an, "BEER: BEtter Evaluation as Ranking," *Proceedings of the Ningth Workshop on Statistical Machine Translation,* pp. 414-419, 2014.

[16] I. Sutskever, O. Vinyals and Q. V. Le, "Sequences to sequences learning with neural networks.," *In Advances in Neural Information Processing Systems,* 2014.

[17] C. Kyunghyun, B. v. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Proceedings of the Empiricial Methods in Natural Language Processing.,* 2014.

[18] K. Moritz Hermann and P. Blunsom, "Multilingual distributed representations without word alignment," *arXiv:1312.6173 Proceedings of the Second International Conference on Learning Representations,* pp. 1-9, 2014.

[19] K. Cho, B. v. Merrienboer, D. Bahdanau and Y. Bengio, "On the properties of neural machine translation: Encoder–Decoder approaches," *In Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation.,* vol. b, 2014.

[20] L. V. S. Chauhan, "Anomaly detection in ECG time signals via deep long short-term memory networks," *Data Science and Advanced Analytics (DSAA). IEEE International Conference on, IEEE,* no. 36678 2015, pp. 1-7, 2015.

[21] T. Mikolov, M. Karaflat, L. Burget and J. Cernocky, "Recurrent neural network based language model," *Eleventh Annual Conference of the International Speech Communication Association,* pp. 1-24, 2010.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, pp. 1735-1780, 1997.

[23] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Gated feedback recurrent neural networks," *arXiv preprint arXiv:1502.02367.*

[24] A. M. Larriba Flor, "Character-based Neural Machine Translation," Master's Thesis. Universidad Politécnica de Valencia, Valencia, 2017.

[25] T. Luong, H. Pham and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* pp. 1412-1421, 2015.

[26] M. Creutz and K. Lagus, "Unsupervised Models for Morpheme Segmentation and Morphology Learning," *Transactions on Speech and Language Processing,* vol. 1, no. 4, pp. 1-34, 2007.

[27] H. Zhao, C.-N. Huang, M. Li and B.-L. Lu, "A unified character-based tagging framework for chinese word segmentation," *Association for Computing Machinery. Transactions on Asian and Low-Resource Language Information Processing,* vol. 2, no. 9, pp. 1-32, 2010.

[28] R. Senrich, B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," *arXiv:1508.07909v5,* 2016.

[29] P. Gage, "A new algorithm for data compression," *The C Users Journal,* pp. 23-38, 1994.

[30] R. Patel, "Iconic Translation," The Neural MT WEELY, December 2019. [Online]. Available: https://iconictranslation.com/2019/12/forward-vs-back-translation-for-neural-mt/. [Accessed 15 November 2020].

[31] N. Bogoychev and R. Sennrich, "Domain, Translationese and Noise in Synthetic Data for Neural Machine Translation," *arXiv:1911.03362v2,* 2020.

[32] M. Graça, Y. Kim, J. Schamper, S. Khadivi and H. Ney, "Generalizing Back-Translation in Neural Machine Translation," *arXiv:1906.07286v1,* 2019.

[33] S. Edunov, M. Ott, M. Auli and D. Grangier, "Understanding back-translation at scale.," *arXiv:1808.09381v2.*

[34] G. Lample, M. Ott, A. Conneau, L. Denoyer and M. Ranzato, "Phrase-Based & Neural Unsupervised Machine Translation," *arXiv:1804.07755v2,* 2018.

[35] J. Gu, Y. Wang, Y. Chen, K. Cho and V. O.K. Li, "Meta-Learning for Low-Resource Neural Machine Translation," *arXiv:1808.08437v1,* 2018.

[36] M. Artetxe and H. Schwenk, "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond," *arXiv:1812.10464v2,* 2018.

[37] D. Stefan Munteanu, A. Fraser and D. Marcu, "Improved machine translation performance via parallel sentence extraction from comparable corpora," *ACL,* 2004.

[38] A. Irvine, "Combining bilingual and comparable corpora for low resource machine translation," 2013.

[39] A. Irvine and C. Callison-Bursh, "End-to-end statistical machine translation with zero or small parallel text.," *Natural Language Engineering,* vol. 1, 2015.

[40] H. Zheng, Y. Cheng and Y. Liu, "Maximum expected likelihood estimation for zero-resource neural machine translation," *IJCAI,* 2017.

[41] P. Koehn., "Europarl: A Parallel Corpus for Statistical Machine Translation," *Proceedings of the Machine Translation Summit,* pp. 79-86, 2005.

[42] G. Klein, Y. Kim, Y. Deng, J. Senellart and A. Rush, "OpenNMT: Open-Source Toolkit for Neural Machine Translation.," *Proceedings of ACL 2017, System Demonstrations.,* pp. 67-72, July 2017.

[43] K. Papineni, S. Roukos, T. Ward and W. Jing Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation.," *Proceedings of the Annual Meeting of the Association for Computational Linguistics,* no. 10.3115/1073083.2073135, pp. 311-318, October 2002.

[44] M. Snover and B. Dorr, "A Study of Translation Edit Rate With Targeted Human Annotation," *in Proceedings of the Association for Machine Translation in the Americas,* pp. 223-231, 2006.

[45] M. Domingo, A. Peris and F. Casacuberta, "Segmented-based interactive-predictive machine translation," *DO: 10.1007/s10590-017-9213-3,* 2018.

[46] X. Garcia, P. Foret, T. Sellam and A. P. Parikh, "A Multilingual View of Unsupervised Machine Translation," *arXiv:2002.02955v4,* 2020.

[47] Y. Park and I. D. Yun, "Comparison of RNN Encoder-Decoder Models for Anomaly Detection," *arXiv:1807.06576v2,* 2018.