Universitat Politècnica de Catalunya
Barcelona Tech

Facultat d'Informàtica de Barcelona (FIB)

Final degree project

Bachelor's degree in Informatics Engineering
(Computer Science Specialization)
Grau en Enginyeria Informàtica (Especialitat en Computació)

Similar language translation

Author:
Pere Vergés Boncompte

Supervisor (Director):
Marta Ruiz Costa-Jussà
Computer Sciences Department

June 21, 2020

Contents

Resum

La traducció automàtica és la tasca de traduir automàticament un idioma a un altre. Aquest projecte avalua el rendiment dels últims sistemes d'aprenentatge profund en la tasca de traducció d'idiomes similars. Avaluarem la traducció entre Català, Castella i Portuguès, que són llengües romàniques, per veure com l'arquitectura del Transformer realitza la tasca. També farem servir diverses tècniques per millorar la traducció entre els idiomes. Primer, utilitzarem model multilingües que permeten fer transferència de coneixement entre idiomes i poder fer traduccions zero-shot. Després aplicarem backtranslation per poder fer ús dels texts monolingües i millorar les traduccions del sistema. Per últim millorarem la traducció de domini específic fent ús de fine tuning.

Resumen

La traducción automática es la tarea de traducir automáticamente un idioma a otro. Este proyecto consiste en evaluar el rendimiento de los últimos sistemas de aprendizaje profundo en la tarea de la traducción de idiomas similares. Evaluaremos la traducción entre el Catalán, el Castellano y el Portugués, que son lenguas románicas, para de esta manera ver como la arquitectura del Transformer realiza dicha tarea. Además usaremos diferentes técnicas para mejorar la traducción entre estos idiomas. Primero, haremos uso de modelos multilingües que permiten transferencia de conocimiento entre idiomas y permiten hacer zero-shot . Después aplicaremos backtranslation para poder usar los textos monolingües y mejorar las traducciones del sistema. Por último mejoraremos las traducciones de domino específico usando fine tuning.

Abstract

Machine translation is the task of automatically translating one language into another. This project aims to evaluate the performance of state-of-the-art Deep Learning systems on similar language translation. We will to evaluate the translation between Catalan, Spanish, and Portuguese, which are Romance languages, and see how the Transformer architecture performs in this task. We will additionally make use of different techniques to improve the translation between these languages. First of all, we will make use of multilingual models that allow for transfer-learning as well as zero-shot translations. Secondly, we will apply the backtranslation technique to make use of the monolingual data and better the system translations. Lastly, we will improve the specific domain data using fine tuning.

Acknowledgments

I am grateful to Marta Ruiz Costa-jussà, supervisor of this project, for giving letting me this research project on Machine Translation with her, and for all the guidance and insight she has given me. I would also like to thank Carlos Escolano Peinado for his comments, corrections and help when debugging the code.

Thanks to UPC-TSC CALCULA clusters all the experiment that we performed has been possible, without them, the research could not have been possible.

1 Introduction

This project aims to participate in the Similar Translation Task proposed by the Fifth conference on machine translation (WMT20)[1]. It is the second time that this task is proposed by the conference.

The Similar Translation Task aims to evaluate the performance of state-of-the-art Deep Learning systems on the translation between languages of the same linguistic family. This year's competition proposed three different language families: Romance, South-Slavic, and Indo-Aryan. In our case, we will only focus on Romance languages, since these are the languages we are more acquainted with.

There are several Romance languages, but the task provides data for Catalan, Spanish and Portuguese, last year's competition proposed the Similar Translation Task only between Spanish and Portuguese. Therefore, we are going to evaluate the Deep Learning system using these three languages. The task itself provides parallel corpus data and monolingual data. Nevertheless, the organizers delivered us with parallel data for the following pairs: Spanish-Catalan and Spanish-Portuguese and monolingual datasets for all the languages. For the evaluation of the systems, we do have parallel domain and test sets.

The idea of our project is to test different models and choose the best ones. However, apart from doing the bilingual translation of the languages which we have parallel data, we will also evaluate multilingual systems for the translation between Spanish, Catalan, and Portuguese.

The area this project focuses on is Neural Machine Translation (NMT). There are various state-of-the-art architectures for NMT. However we are going to use one architecture that uses attention layers. This architecture is called the Transformer [6], and we are going to use it for the training of our models. For implementation we use the Fairseq library[1], which is built on top of Pytorch[2], which uses Python[3] as a programming language. This library is also going to help us with all the preprocessing and evaluation of the data.

To have a wider introduction and the scheduling of this project, there is the GEP document in the Annex B.

Contribution This project contributes to the evaluation of multilingual models on similar languages. With this study we aim to support the improvement of state-of-the-art multilingual systems, and have an overview of how zero-shot works in this kind of systems.

[1] https://Fairseq.readthedocs.io
[2] https://pytorch.org/
[3] https://www.python.org/

We are also going to apply two techniques in order to see how these models perform with them. First we have backtranslation which allows us to introduce monolingual data to the NMT system. The second technique is fine tuning which will improve the translation for the in-domain data.

This project will be participating at the Fifth Conference on Machine Translation (WMT20), and we will publish a paper that is going to be directly accepted with the results that we reach. In the Annex A, there is a draft of the paper that we will deliver on August 15th of 2020.

2 Background

In this section we will explain all the basic concepts needed to understand the project.

2.1 Natural Language Processing (NLP)

Natural Language Processing is a branch of Deep Learning which is itself a branch of Machine Learning methodology that relies on artificial neural networks. These networks want to simulate the brain and reproduce it at a higher level. To reproduce this behaviour, these networks are made of nodes, and each node is itself a logistic regression classifier.

NLP is the study and processing of large amounts of Natural Language data. Therefore, the idea behind it is to work with large amounts of information and automatically get results and information from it. This can be used to translate automatically one language to another, to analyze the sentiment of a sentence, to predict answers to questions, to generate natural language, etc. This discipline is divided in several fields which all have applications to real problems.

In our project, we will be focusing on the Machine Translation field, which aims to automatically translate sentences from one language to another.

2.2 Machine Translation (MT)

Machine Translation is the area of Natural Language Processing whose objective is to automatically translate one natural language to another while preserving the meaning and intention of the text. It is basically a Language Modelling that is conditioned to a source sentence[4].

There are different models of Machine Translation. One of the most famous ones is seq2seq modeling, which consists of two Recurrent Neural Networks (RNN); one is an encoder and, the other one is a decoder. This model works by firstly, entering a sequence of words (a sentence) and then, retrieving another sequence of words (output) which is the translation of the first one. The problem with these models is that they usually do not perform well when treating long sentences. Thus, we are using a new state-of-the-art-architecture which is based on the Transformer[6].

Transformer uses attention to boost the speed with which Neural Machine translation models are trained [5].

Transformer has two big components. One is the encoder which is itself composed by a set of encoders. And the other, the decoder, composed itself by a set decoders. It is important to note that they both must be composed by the same number of encoders/decoders.

Each encoder is made of two components. One is the self-attention layer (this layer helps the encoder look at other words in the input sentence as it encodes a specific word) and, the other is a feed-forward Neural Network. The decoder looks quite alike but it has an additional masking that forces to put attention only to previous tokens.

Each input word is represented as a vector using an embedding algorithm and flows through each of the two layers of the encoder. Each word follows its own path in the encoder, which means that it can be executed in parallel.

The dependencies between words are in the self-attention layer (which helps to acquire the understanding of other relevant words to the one we are processing). In Figure 1 we can see a representation of the Transformers architecture.

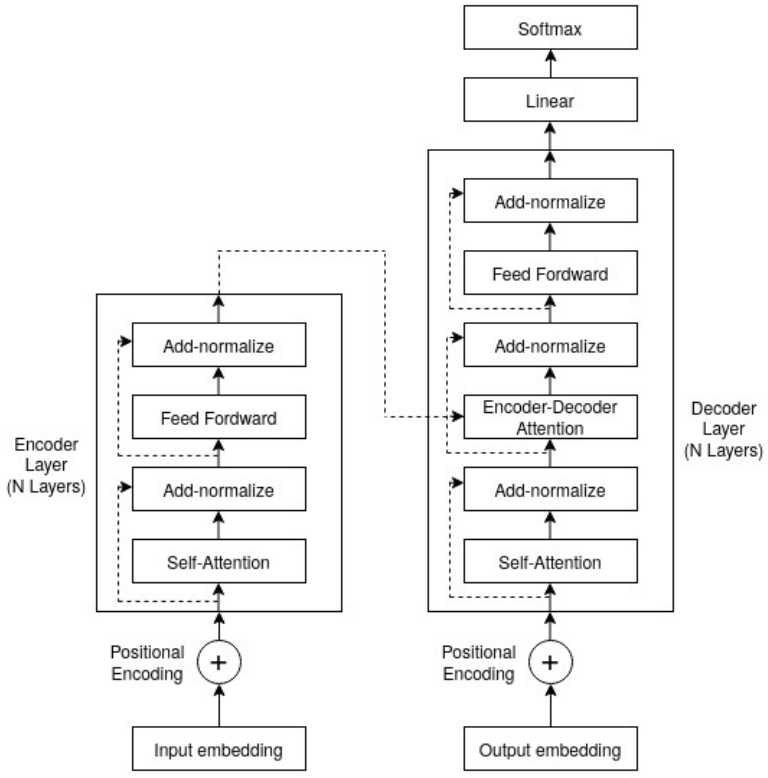

Figure 1: Transformer architecture representation, based on Figure 1 in [6].

Self-Attention It supports the creation of a relationship for each word with all the other words in the sentence, in Figure 2 is shown a representation of this relationship works. This helps the enrichment of the encoding for each word since it will take into account this information to infer meaning to the word. Now, we will explain how self-attention works.

For each word, we create three vectors: a Query vector, a Key vector, and Value vector. These vectors are created by multiplying the embedding by three matrices that we trained. Their dimensionality is 64, whereas the inputs vectors are 512.

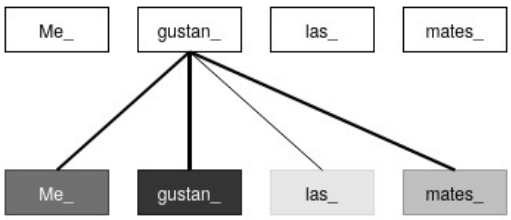

Figure 2: Example of how self-attention works. Using self-attention the model is aware of all the input embeddings for each word position. The darker the background of the word is, the more relevant the word is for the word the model is treating at the moment.

Then, we need to calculate the score. To do this we need the score of each word in the sentence against the one we want to calculate, that is the one we are treating at the moment. This score will give us the amount of focus and attention we have to give to each word in the sentence against the word in question. This score is the result of the dot product of the query vector and the key vector of the word we are treating. Then, we divide the scores by the square root of the dimension and send this result to a soft-max operation to normalize it. Afterwards, we have to multiply each value vector by the soft-max score to avoid the noise of irrelevant words. The last step is to sum the weighted vectors. The result is the output of the self-attention layer.

2.3 Multilingual architecture

In the following section, we will explain the architecture used to evaluate the similar language translation task.

Shared This architecture was developed by Google and showed a substantial improvement in comparison to the previous models. It is based on the Transformer and it only uses attention mechanisms.

An important aspect of this architecture is that helps low-resource language

to achieve improvement of all language translations in the model. This is because it forces to generalize across all languages during training, and it will help low resource languages in the model to reach improvement because of the positive transfer learning [19].

Moreover, the shared architecture allows the use of the zero-shot technique in an easy manner, since we can use a single Neural Machine Translation model to translate between all languages directions, just by adding a tag before the input sentence. The feature this architecture has to enable the zero-shot transfer learning is going to be key to generate translations between Portuguese and Catalan.

Zero-shot is a technique that is based on inference of knowledge of a supervised system to classes that do not have labeled data. In our case, we have a system of three languages in which two out of three have labeled data. So, we are going to apply zero-shot, to infer the necessary knowledge to translate from Catalan to Portuguese and from Portuguese to Catalan. In Figure 3 there is a representation of how zero-shot will work in our case.

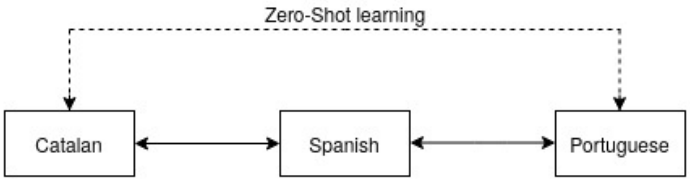

Figure 3: Zero-shot representation for our multilingual system, set with parallel data for the directions of Catalan-Spanish and Spanish-Portuguese. The translation between Catalan-Portuguese are inferred with zero-shot

2.4 Backtranslation

To improve the translation of our models trained with bilingual data, we are going to use the backtranslation technique. This technique consists of translating a monolingual corpus (mono.src) using the src-tgt model previously trained and getting the translation (mono.tgt). With this translation, we have achieved a new "bilingual" corpus, since we have a source and target corpus that have the same sentences. We will add these data to the previous training corpus and retrain the model from scratch [12]. In Figure 4, we show a representation of the backtranslation flow for the Catalan addition to the Spanish to Catalan direction.

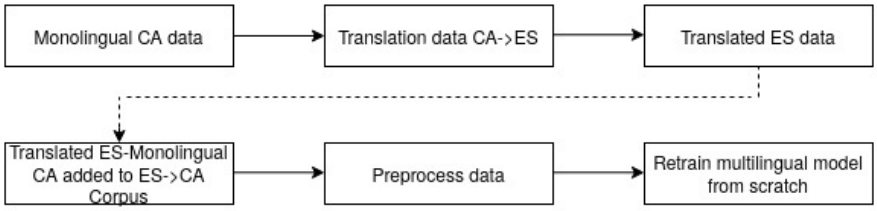

Figure 4: Backtranslation process example for the ES-CA direction.

Using backtranslation we should perceive an improvement of the newly trained model as compared to the previous one, even taking into account that the translation of the mono.tgt is not absolutely correct.

We were given a lot of monolingual data. For this reason, we were not able to use all of it for backtranslation since the process of translating sentences takes a lot of time. Thus, we needed to choose the best documents to translate from; to do this we decided to compare the corpus of the monolingual documents against the domain sets.

TF-IDF Which means Term Frequency - Inverse Document Frequency, is used to see the relevance of words in a document.

We decided to use this method to compare the monolingual data we had, and we used a Github script (*ranker.py*[4]) that compared the similarity of documents using this criterion. We picked the documents that had a higher score of similarity.

2.5 Domain adaptation: Fine tuning

Another way to improve models is to use fine tuning. This method consists of adding in-domain data to the corpus, which is data of the domain we want to translate. Adding this data is going to improve the translation of the specific domain you add, to the already trained model. So instead of generalizing the model you are training, the target of this method is to make the model improve for a specific domain. One downside that this method can have is that the model over-fits easily.

[4]https://github.com/BhargavaRamM/Document-Similarity

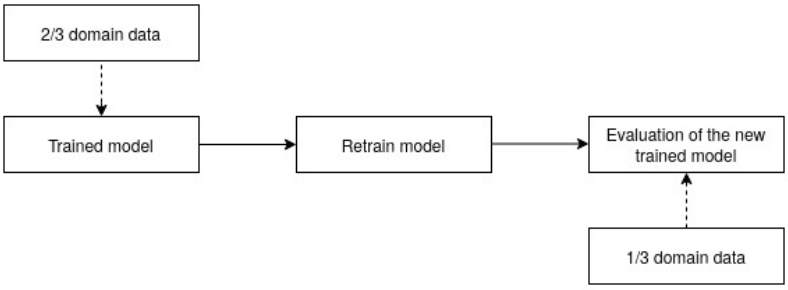

Figure 5: Fine tuning process representation.

In our case, we want to have a better translation of our domain test like data, so we are going to split the domain set and add it to the corpus [13]. In Figure 5 is shown the process we will follow to do the fine tuning. Specifically, we are going to split the domain set in 2/3 for the fine tuning and 1/3 for the evaluation, using it as test data. When comparing this model with the other ones we will have to evaluate the other models with the same 1/3 of domain data we used for the evaluation of the fine tuned models.

2.6 Data Preparation

For Deep Learning, it is essential to have a good set of data. One key element to achieve this is to prepare the data the right way, and get rid of irregularities in it. In the case of Machine Translation, to prepare the data you need to do a series of operations to clean it and set it for the training.

There are several steps to follow when preparing and cleaning the data. Now we will explain which steps we will take to prepare the data. Keep in mind that some of these steps are performed only because we are using Fairseq directives which require to prepare the data in a certain way. To do most of the preparation, we will be using scripts from the Moses[5] decoder and Subword NMT[6].

Normalize punctuation The first step to prepare the data is normalizing punctuation. This means to remove and standardize it, which is removing extra spaces and dealing with pseudo-spaces. Moreover, we must normalize all the Unicode punctuation marks and all the quotation marks. This first step will be performed by using the *normalize-punctuation.perl*, from the Moses decoder scripts.

[5]https://github.com/moses-smt/mosesdecoder
[6]https://github.com/rsennrich/subword-nmt

Remove non-printing characters In this step we are removing all the non-printing characters that we did not remove with the normalized punctuation step. These characters are, for instance: space, non-breaking space, pilcrow, tab character... As we did in the previous step we are using a Moses decoder script to do this task, which is the *remove-non-printing-char.perl*.

Tokenize This is a very important step, once all the irrelevant characters have been cleaned from all the data, we have to tokenize the data [8]. Tokenizing the data means to separate the data into smaller pieces, it usually means splitting the input sentences into words and important characters. When doing this process it is very important to know in which language we are tokenizing, since there are specific patterns for each language that are important to take into account.

To illustrate this process, we are going to carry out a simple example of the result of tokenizing a sentence:

How are you doing?

After tokenizing this sentence, the result would be an array such as:

[*How*, *are*, *you*, *doing*, *?*]

As we have done in the previous steps of cleaning the data, we will be using one of the Moses decoder scripts. This script is the *tokenizer.perl*.

Cleaning data This step of data preparation consists in deleting all the sentences that do not apply a certain criteria. The criteria normally used in this pruning of the data are the length of the sentences and the length difference of the same sentence from one language and the other language, if there are empty sentences, the maximum number of characters of a word among others.

This is a very important step since it helps enormously to prune the data and delete obvious bad translations from the input data. This is the reason why this step is one of the most important ones for improving the performance of the model.

For this step, we will be using the *clean-corpus-n.perl* from the Moses decoder scripts.

Truecase Once we have tokenized the data we can then start proceeding with the truecase step. This procedure consists of lower-casing all the data except the words that start with uppercase, these words are usually names of people, cities and so on. So this process will lowercase virtually all the first characters of every sentence and all the words that are not always written with a starting uppercase. This step is not going to improve the performance of the model greatly, but it will help its performance.

This procedure is going to be done using the *truecase.perl* script from Moses decoder [9]. Before using this script, it is necessary to train the truecaser with the data, to learn the words that are needed to preserve the starting uppercase.

Byte pair encoding (BPE) BPE is a data compression algorithm that compresses pairs of common characters and replaces them by a single character that does not appear in the data. For example, you could substitute the pair "ou" for the single letter "A", if "A" did not appear in your data [10].

Using this technique we can reduce the vocabulary of our data, only increasing minimally the number of tokens used to encode the vocabulary.

To do this procedure, we are going to use the *learn_ bpe.py* and *apply_ bpe.py* scripts from the Subword NMT scripts. Firstly, we need to learn the possible encoding of the training data and then, we will apply this knowledge to all the data.

Shared Multilingual data preparation For this task we are going to do a similar preparation but with some tweaks, we are going to follow all the steps we did before for the four directions that we have data, but before doing the BPE step we need to add special tags to each sentence of each file.

This is the configuration of dataset we are going to use and the tags we are going to add to each sentence.

SRC: *<2es> .ca*	***TGT***: *.es*	*From ES-CA dataset*
SRC: *<2ca> .es*	***TGT***: *.ca*	*From ES-CA dataset*
SRC: *<2pt> .es*	***TGT***: *.pt*	*From ES-PT dataset*

This configuration is going to be applied to each training, validation and testing set of the specified ones, and after applying these tags we are going to concatenate all the training, validation and testing sets in one file, one for the SRC which will be .src and one for the TGT which will be .tgt. Once we have concatenated all the files then we are going to shuffle the data using a random seed, these shuffling is done by Fairseq, but only in case it does not do it we are going to shuffle the data first. We are doing this process because we do not want that in one batch it only finds sentences from one language.

Now that we have all the data with its tags and shuffled we are going to train the BPE with the training data, we are using the train.src and train.tgt concatenated to train the BPE and once it is trained we are going to apply it to the training, validation and testing sets.

2.7 Evaluation

For Machine Translation there are several methods to evaluate the translations. This metrics evaluates the quality of the translations performed by the system. Now, We will make a brief explanation of the most used methods:

- **Word error rate** This method consist of the calculation of the number of words translated that differ from the reference translation. It is also based on the Levenshtein distance which works at a character level.
- **National Institute of Standards and Technology (NIST)** This method is based on the BLEU scoring metric, but with some modifications. A part from basing the calculation on the n-grams (are a sequence of n items, which can be words, characters, etc) precision it also adds a weight factor for each n-gram, the more different the n-gram is, the more weight will be assigned to it.
- **Metric for Evaluation of Translation with Explicit ORdering (METEOR)** This metric was designed to fix some of the problems of the BLEU scoring metric. It consist on the harmonic mean of the unigram, precision and recall of the translation.

The method that we will be using is the Bilingual Evaluation Understudy (BLEU), this is because the competition is going to use this scoring methods as well as human evaluation.

BLEU The BLEU metric [17] correlates highly with human evaluation, and the idea behind it is to evaluate a sentence comparing it to a professional human translation, so the closer to this translation it is the higher the score it gets.

BLEU is very similar to precision, which in Machine Translation the precision is between ngrams, but it makes some modifications able to compare it to multiple references

$$P = \frac{common\, ngrams}{reference\, ngrams}$$

The modification of the precision is done by adding a penalty for length difference between translations, because otherwise a sentence that has two words that are very common such as the, as, be, and a long sentence that contains such words would have a precision of 1[31]. This modification works as this:

$$PB = \begin{cases} 1 & \text{if c} > \text{n} \\ e^1 - \frac{r}{c} & \text{Otherwise} \end{cases}$$

So the formula to calculate the BLEU score is the following:

$$BLEU = PB \times exp(\sum_{n=1}^{N} w_i \, log P_n)$$

Evaluation process Once the training of the model has finished we have different checkpoints of the model. To evaluate it we are going to pick the best checkpoint. If we have enabled the bleu-score parameter, we will use it to pick the best one; if we can not use this parameter, the best model is going to be the one that has the lowest loss.

Then, to evaluate the model we could use the *Fairseq-score*, which calculates the BLEU score of generated translation against its references. But instead of using the *multi-bleu.perl* script from the Moses decoder, because we will need it when we do the multiple translations, and in case we need to evaluate a different domain or test set.

To use this script to evaluate the model we will first have to apply the tokenization, the truecasing and lastly the BPE, to the input test data. For the BPE we will use the trained BPE of the train data of the language we are translating. Once we have applied the BPE we are going to use the command-line *Fairseq-interactive* to translate the test data of the source language. This will generate a document with the data translated. However, we have to clean the document to only have the translated output. Once we have the document cleaned, we can apply the *multi-bleu.perl* script to it and get the score of the translation of the test data against the target test set data.

3 Similar Language Translation

As we have explained in the introduction, this project consists in using Machine Translation with the Transformer architecture to translate languages from the same linguistic family. Our main purpose is to translate between Spanish, Catalan, and Portuguese with the data proportioned by the WMT20. The data that we have been given is composed of two parallel data sets one Spanish and Catalan and the other Spanish and Portuguese. We have also been given 29 monolingual documents in total for all three languages.

We will use two different model architectures, one bilingual and one multilingual.

The approach we are going to take is very similar for all of three models and it consist of four steps:

- **First step: Prepare data.** In this step, we will clean and prepare all the input data to allow their preprocess.
- **Second step: Preprocess data.** Once we have the data prepared, we can now use the fairseq-preprocess command line to create the dictionary and binarize the data.
- **Third step: Train data.** This step, is the one that will take more time. Once we have preprocessed the data we will make use of the fairseq-train command line to train the model.
- **Fourth step: Evaluation of the data.** Using the models trained in the previous step, we are going to evaluate their performance using the test set, in our case, we will use the domain set, which is 1/3 of the domain set given to us by the competition.

3.1 Data

First of all we will introduce the data that we have been given by the WMT20. This is the data that we will use to train and evaluate all of our models.

The most important data, to train our models, is the parallel corpus data, which consists of two documents written in different languages that are composed of the same sentences in the same order. In our case we have one parallel set corresponding to the Spanish and Catalan directions and the other one for the Spanish and Portuguese, we do not have bilingual data for the Catalan and Portuguese directions.

The other data that we were given is a great number of monolingual documents of every language we need to evaluate.

3.1.1 Parallel data

Catalan and Spanish dataset The parallel corpus for the Catalan-Spanish directions are composed of two documents.

The first document is a set of parallel Wikipedia titles, it has exactly 446.326 titles. Whereas the second file is a huge number of documents from the journal of the Catalan Government, it has 10.933.622 sentences. From these files, we will generate the training and validation sets. Table 1 summarizes the sentences and tokens each data set has.

Corpus name	**#Sentence**	**#Tokens**
Wikititles-v2	ES: 446.326 CA: 446.326	ES: 1.035.601 CA: 1.045.310
DOGC-v2	ES: 10.933.622 CA: 10.933.622	ES: 150.435.197 CA: 162.981.769
Total	ES: 11.379.948 CA: 11.379.948	ES: 151.470.798 CA: 164.027.079

Table 1: Catalan and Spanish bilingual corpus dataset.

Spanish and Portuguese dataset The parallel data set we have is composed of four documents. In this case, we have more documents but less data (number of sentences), that we have for the Spanish-Catalan directions.

The first document we have is the Europarl-v10, this document is formed of 1.801.845 sentences, the corpus data of this document is extracted from the European Parliament proceedings. Then we have the News-Commentary-v15. This one has 48.259 sentences, quite a small file compared to the other ones, this is a news commentary corpus, which is a generic news document. The third document is like the one we had in the Spanish and Catalan dataset, a Wikipedia titles document, wikititles-v2, in this case, it has 649.833 number of sentences. The last one is the JRC-Acquism, its corpus has a total of 1.650.126 sentences, this document is a collection of Europian Union legislative texts. Table 2 shows these files with their number of sentences and tokens.

Corpus name	**#Sentence**	**#Tokens**
Europarl-v10	ES: 1.801.845 PT: 1.801.845	ES: 47.832.341 PT: 46.191.464
News-Commentary-v15	ES: 48.259 PT: 48.259	ES: 1.273.308 PT: 1.220.832
Wikititles-v2	ES: 649.833 PT: 649.833	ES: 1.649.959 PT: 1.618.352
JRC-Acquis	ES: 1.650.126 PT: 1.650.126	ES: 35.868.080 PT: 33.474.269
Total	ES: 4.150.063 PT: 4.150.063	ES: 86.623.688 PT: 82.504.917

Table 2: Portuguese and Spanish bilingual corpus dataset.

3.1.2 Monolingual data

The WMT20 has provided us with a lot of monolingual data, in total 29 documents, we have 14 documents that are in Spanish, another 14 in Portuguese and one document in Catalan, even though this last one is formed of 24 million sentences.

The data shown in Table 5 is aimed to be used for backtranslation purposes. We are going to make the backtranslation process for the best model or models we have, because this is a very expensive and slow process.

Backtranslation The documents shown in Table 5 are the documents that we are going to use for backtranslation. We have chosen the news-commentary datasets from all the ones we had of Spanish and Portuguese since they are generic and of high quality. We will use the CaWaC document for the Catalan backtranslation, since is the only one we have. In Table 3 we show its TF-IDF score compared against the domain set.

Corpus name	**Similarity score**
News-commentary ES	0.9836836325152
News-commentary PT	0.9645670084526
CaWac CA	0.9592066341636

Table 3: List of backtranslation documents, with their TF-IDF score (0 to 1)

Then, for the Spanish and Portuguese backtranslation we decided to add an additionally document from the News-Crawl datasets, because the other document has a very specific domain (a document extracted from the European Parliament proceeding) and it is not going to generalize as well as it will do with the News-Crawls.

To do the selection of all the documents we will compare the resemblance between them using the TF-IDF metric, which is the Inverse Document Frequency, and see which one is more similar to the domain set. We will be using an script from *BhargavaRamM*, to do the TF-IDF evaluation of the document similarity [14].

NewsCrawl Spanish and Portuguese datasets It is shown in Table 4 the scores each document has achieved using the TF-IDF evaluation.

Corpus name	**Similarity ES**	**Similarity PT**
News-crawl-2008	0.9839725018543	0.9689088564309
News-crawl-2009	0.9860422585384	0.9762894348406
News-crawl-2010	0.9862083349540	0.9783696388370
News-crawl-2011	0.9854518989870	0.9773981819003
News-crawl-2012	0.9854838965793	0.9752528652577
News-crawl-2013	0.9851074086928	0.9796982342588
News-crawl-2014	0.9850937925499	0.9787888993599
News-crawl-2015	**0.9864468697037**	0.9795708278473
News-crawl-2016	0.9850178696991	0.9791399943303
News-crawl-2017	0.9856714146459	0.9787694392134
News-crawl-2018	0.9841282698524	0.9818951893442
News-crawl-2019	0.9838691708624	**0.9823996499398**

Table 4: List of Spanish and Portuguese News-Crawl backtranslation documents, with their TF-IDF score (0 to 1)

As we show in Table 4, we are going to be using the 2015 News-Crawl for the starting Spanish directions, but not all the dataset since it has three million sentences. We will translate 997.100 sentences to Portuguese and 947.432 sentences to Catalan. For the Portuguese starting directions, we will use the 2019 News-Crawl translating 625.865 sentences to Spanish and 740.564 to Catalan.

For the Catalan dataset, since it is huge, we are only going to translate a certain number of sentences due to time constraints. We have translated 1.923.759 sentences for CA→ES direction and 2.263.212 sentences for CA→PT directions.

Corpus name	#Sentence	#Tokens
News-commentary-v15	ES: 465.165	11.731.242
News-crawl-2015	ES: 3.870.691	109.001.929
News-commentary-v15	PT: 73.550	1.782.489
News-crawl-2019	PT: 1.091.038	21.240.641
CaWaC	CA: 24.745.986	733.974.712

Table 5: Corpus used for the backtranslation

3.2 Data Preprocessing

Once we have prepared the data using the procedure explained in the background section, we need to binarize it and generate the dictionaries to be able to train the model using Fairseq.

For this step, we will use the *Fairseq-preprocess* command line. We will assign to the train, validation, and test for the preprocessing the splits we did when preparing the data. For each model we will set the corresponding source and target languages and we will use the joined-dictionary flag, this is going to share the embedding and make the languages in the model share the same dictionary.

3.2.1 Bilingual model

In Table 6 we show the preprocessing of the bilingual data.

Model	Lang	%Train rep	%Valid rep	Dict type
CA-ES	CA	0.0	0.000244	40.519
	ES	0.0	0.000171	40.519
ES-CA	ES	0.0	0.000171	40.519
	CA	0.0	0.000244	40.519
PT-ES	PT	0.0	0.000293	40.759
	ES	0.0	0.000290	40.759
ES-PT	ES	0.0	0.000290	40.759
	PT	0.0	0.000293	40.759

Table 6: Data preprocessing for all four directions of ES-CA and ES-PT, it is shown the number of tokens replaced by unknown for each train and valid sets.

3.2.2 Shared Multilingual model

In the multilingual model since we are doing the zero-shot we only have one source language, which is the combination with tags of the three we are using, and one target language.

Multilingual In Figure 7 we see the preprocessing result of the multilingual model.

Lang	%Train rep	%Valid rep	Dict type
SRC	0.0	4.3e-05	40.983
TGT	0.0	4.6e-05	40.983

Table 7: Data preprocessing of multilingual model, it is shown the number of tokens replaced by unknown for each train and valid sets.

Backtranslation In Figure 8 we see the preprocessing result of the multilingual model.

Lang	%Train rep	%Valid rep	Dict type
SRC	0.0	1.12e-05	41.463
TGT	0.0	1.18e-05	41.463

Table 8: Data preprocessing of multilingual model with backtranslation, it is shown the number of tokens replaced by unknown for each train and valid sets.

Fine tuning multilingual with backtranslation For the preprocessing of the fine tuning, we will add a couple of flags to have the same dictionaries that we used in the previous model, so we can retrain from the last-checkpoint.

–srcdict
–tgtdict

In Table 9 we see the result of the preprocessing for the multilingual model with fine tuning.

Lang	%Train rep	%Valid rep	Dict type
SRC	0.00159	0.00174	41.463
TGT	0.00168	0.00183	41.463

Table 9: Data preprocessing of multilingual model of backtranslation with fine tuning. It is shown the number of tokens replaced by unknown for each train and valid sets.

3.3 Parameters

In this section we will show which configuration we will be using in the models, this configuration is the one that we will put in the fariseq-train command line.

Bilingual and Shared Multilingual For these models we are going to be using virtually the same configuration, except for the *share-all-embeddings* parameters added for the multilingual. This parameter is going to allow to share the embedding weights between source and target before the softmax function.

Configuration	**Parameter**
arch	transformer_iwslt_de_en
share-decoder-input-output-embed	
optimizer	adam
adam-betas	(0.9,0.98)
clip-norm	0.0
lr	0.0001
lr-scheduler	inverse_sqrt
warmup-updates	4000
warmup-init-lr	1e-07
dropout	0.1
weight-decay	0.0
criterion	label_smoothed_cross_entropy
label-smoothing	0.1
max-tokens	4096
update-freq	8
eval-bleu	
eval-bleu-args	beam:5,max_len_a:1.2,max_len_b:10
eval-bleu-detok	moses
best-checkpoint-metric	bleu
maximize-best-checkpoint-metric	

Table 10: Training for the bilingual and shared multilingual models.

The Transformer architecture that we are using is already implemented in Fairseq. The implementation was used to evaluate the translation between German and English, which are Germanic languages, this means that we are using an architecture that was designed to translate between similar languages, even though the languages are not Romance, this architecture will be useful for our purpose.

Now we will explain a few of the most important parameters we will use for the training of the models:

The *share-decoder-input-output-embed* will help make faster and better translation since it helps the classifier to have better prior knowledge of words and translations. We can use this option since the input and output dimensions have the same size.

The *label_ smoothed_ cross_ entropy* configuration is going to make that our cross-entropy loss function avoids assigning true labels when the classification is expected to be true. This will make that when we classify one word we do not do this with total confidence, which will help our model to avoid over-fitting and overconfidence.

Then we have the *max-tokens* and *update-freq*, in the first instance, we assigned the *max-tokens* values to 4096 but this caused memory problems when running the script in the GPU, so we solved this problem dividing the *max-tokens* by two and increasing the *update-freq* to 2.

The last configuration is the BLEU evaluation, in the statement of the task, it is said that the evaluation of the models will be the BLEU and human evaluation. So, to have a reference on how the model performs, we added this parameter that evaluates the model at every epoch, which is a full cycle of training the data set.

3.4 Hardware

All the experiments we have done, have been executed on an NVIDIA TITAN X GPU. This GPU not only has been used to train the models, but also we have used it to do all the translations for the backtranslation process.

Table 11 shows the time taken for each model we have trained in days.

Model	**Training days**
Bilingual ES-CA	5
Bilingual CA-ES	5
Bilingual ES-PT	6
Bilingual PT-ES	5
Multilingual	7
Multilingual with backtranslation	10
Multilingual with backtranslation with fine tuning	3

Table 11: This table shows the number of days each model took to train.

4 Results

Using the parameters we have established in the previous section, we will train the models until they stop improving. We can see if they stop improving using two parameters, the loss or the BLEU score, thanks to the eval-bleu parameters we have added in the training scripts.

After having the models trained, to know how well do they perform in our domain. To do such a task, we will be using the domain set that were given to us by the WMT20. As mentioned in the background section, we have the *Fairseq-interactive* command, that let us make this evaluation. We need to provide the function with the source domain set, the target domain set and the checkpoint of the model we want to evaluate the performance with. This will give us the BLEU score for the translation of the domain sets.

First of all, we will compare the results of the bilingual and multilingual models and see which of the two has a better performance. From the result, we will pick the better one and start doing improvements to it, in order to have a better performance.

For all the figures shown in this section we are generating the CA→PT and PT→CA directions by means of pivot. Which means, for instance of the CA→PT directions, translating first the domain set for the CA→ES direction and then translating the result using the ES→PT model. We decided to do it this way in order to be able to compare the CA→PT and PT→CA directions for all the models since, for the bilingual data we can not translate these directions directly with our model. For the multilingual models, we will be reporting the result of the model for the CA-PT directions, but it will not be shown on the figures.

4.1 Bilingual vs Multilingual

Bilingual For the bilingual we have an overall training score of 91 BLEU for the CA-ES models and a 52 BLEU score for the PT-ES one.

Multilingual For the multilingual model we have obtained an overall training score of 82.44 BLEU.

Now we are going to show the result of the evaluation for both models on Figure 6. For the zero-shot multilingual we have an score for the CA→PT direction of 12.47 and for the PT→CA direction of 17.67 BLEU, which is much less than the pivot strategy. However, when translating with zero-shot we are doing only one translation.

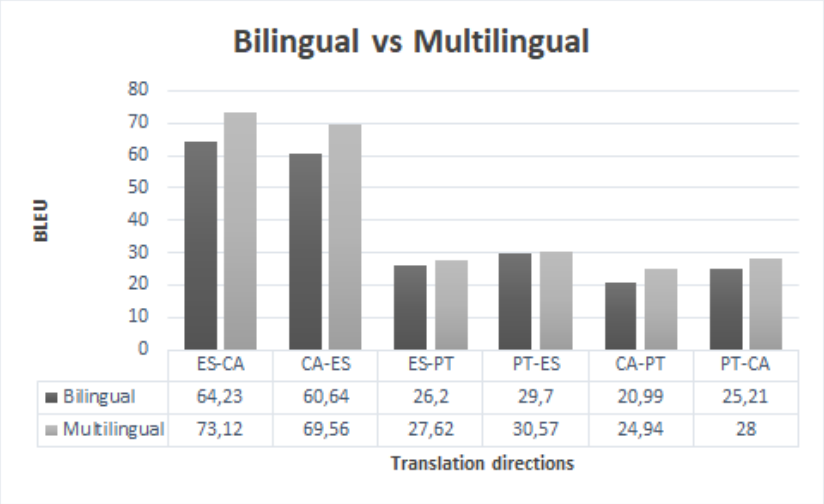

Figure 6: Bilingual vs Shared multilingual BLEU scores.

The results show quite a huge improvement in the CA-ES directions, around a BLEU of +9 BLEU, and for the PT-ES directions we do not see this big of an improvement, it is only around +1 BLEU. For the CA-PT directions we see quite a big improvement for both directions, +4 BLEU for the CA→PT and +3 for the PT→CA, keep in mind that for the CA-PT the scores reported are used using the pivot method.

After this result we see how the multilingual model is better than the bilingual, so for the improvements, we are only going to make them with the multilingual, since we only want to spend resources on bettering the best model.

In Table 12 we see a sample translation for both models and all the directions. In this table, we can discern the difference of learning from the bilingual

model compared to the multilingual one. In the sentence for the ES→CA direction we can appreciate how the bilingual model has not learned that in Catalan, before a vocal (or "h" + vocal) the article "de" contracts to "d'".

System	**Sentence**
ES→CA	
REF	La resposta d'Herrera la va decebre
BILINGUAL	La resposta de Herrera la decepcionó
MULTILINGUAL	La resposta d'Herrera la va decepcionar
CA→ES	
REF	¿Qué es lo que enciende el conflicto?
BILINGUAL	¿Qué es el que encién el conflicto?
MULTILINGUAL	¿Qué es lo que enciende el conflicto?
ES→PT	
REF	sites de saúde feminina estão bloqueados.
BILINGUAL	os sítios de saúde das mulheres estão congelados.
MULTILINGUAL	os sítios de saúde feminina estão bloqueados.
PT→ES	
REF	Reproduzca y organice su colección multimedia
BILINGUAL	Reproducciones y organización de su colección multimídia
MULTILINGUAL	Reproduzca y organice su colección multimedia
CA→PT	
REF	Era o fim da tarde.
BILINGUAL	Era o fim da noite Prada .
MULTILINGUAL	Era o fim da tarde.
PT→CA	
REF	El dispositiu de so s'ha desconectat.
BILINGUAL	El dispositiu de so es va desconnectar.
MULTILINGUAL	El dispositiu de so s'ha desconectat.

Table 12: Translation examples comparison between reference sentence, bilingual model and multilingual model.

4.2 Multilingual vs Multilingual with backtranslation

The data used for this backtranslation is the one stated in the shared multilingual model with backtranslation. We have obtained an overall score of 81.59 for the validation set. Figure 7 shows the scores for all directions, for both multilingual and multilingual with backtranslation. For the CA-PT directions translated directly using zero-shot we obtained, for the CA→PT direction an score of 12.47 for the multilingual model, whereas for the multilingual with backtranslation an score of 13.56. For the PT→CA direction an score of 17.67 and for the multilingual and 19.64 for the backtranslation.

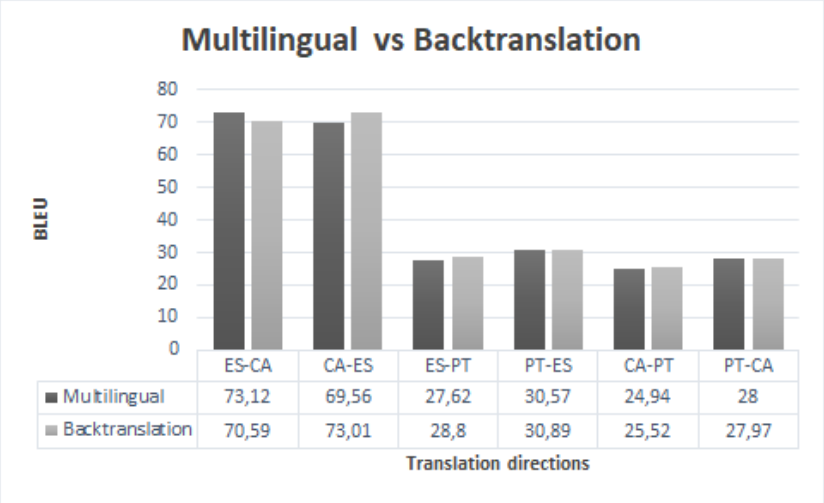

Figure 7: Bilingual vs Shared multilingual BLEU scores using pivot for Portuguese-Catalan direction.

The backtranslation in this multilingual model has improved every direction except for the ES-CA and the PT-CA, but this last one only worsens for 0.03. The other directions achieve an improvement of between +0.3 to +1 for all of them except for the CA-ES direction which reaches a +3 BLEU improvement.

In Table 14 are shown a sample translation of both models for every direction. We see how the model has learned the order of the components of a sentence. This behaviour can appreciated for the CA→ES sentence and also for the PT→CA.

System	Sentence
ES→CA	
REF	Recorda que la teva idea t'està esperant.
MULTILINGUAL	Recorda que el teu idea te està esperant.
BACKTRANSLATION	Recorda que la teva idea t'està esperant.
CA→ES	
REF	¿Los hackers dan problema?
MULTILINGUAL	¿Dan los hackers problema?
BACKTRANSLATION	¿Los hackers dan problema?
ES→PT	
REF	O perfil não está em conformidade com o protocolo
MULTILINGUAL	O perfil não é conforme com o protocolo
BACKTRANSLATION	O perfil não está em conformidade com o protocolo
PT→ES	
REF	¡Finaliza tu compra y aprovecha el descuento!
MULTILINGUAL	¡Acaban sus compras y se aprovechan de los descuentos!
BACKTRANSLATION	¡Finaliza su compra y aprovecha el descuento!
CA→PT	
REF	Informação do Sistema
MULTILINGUAL	Informações sobre o sistema
BACKTRANSLATION	Informação do sistema
PT→CA	
REF	Pot un blog ser considerat literatura?
MULTILINGUAL	Es pot considerar un bloc literatura?
BACKTRANSLATION	Pot un blog ser considerat literatura?

Table 13: Translation examples comparison between reference sentence, multilingual model and backtranslation model.

4.3 Backtranslation vs Backtranslation with fine tuning

With the results we have obtained from the previous models, we have observed how the multilingual with backtranslation had a better score than the multilingual without backtranslation, between +1 and +3 BLEU improvement. That is why we have decided to apply fine tuning to the model with backtranslation, in order to improve the translation data of the domain set.

For the backtranslated with fine tuning model we have an overall score of 81.83 BLEU, which slightly improves the score we had for the model without fine tuning, for 0.24 BLEU.

The scores of both models are shown in Figure 8. As we did for the previous model, now we will state the scores for the CA-PT directions evaluated directly by the models without using pivot. The CA→PT direction we have 13.56 BLEU for the backtranslated and 16.05 for the fine tuned, we see an improvement of +2 BLEU. For the PT→CA we have 19.64, for the backtranslated and 19.56, for the fine tuned an improvement of +0.5.

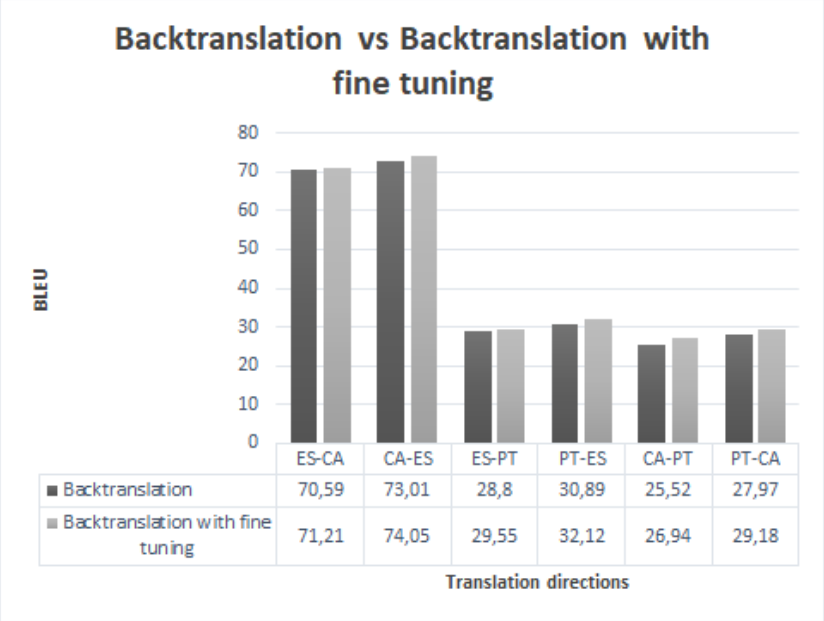

Figure 8: Shared multilingual with backtranslation vs shared multilingual with backtranslation with fine tuning BLEU scores using pivot for Portuguese-Catalan direction.

We see an improvement of +1 for all directions. Using fine tuning we should observe an improvement in domain of the vocabulary of the system translation. We can appreciate these behaviour in Table 14 for ES→CA direction. The

model has learned the word "Moga" whereas before used the word "Mova", and for the ES-PT direction we see it with the words "Remover" and "paus", which before used "Retirar" and "trébolas".

System	**Sentence**
ES→CA	
REF	Moga el sis de trévols al cinc de trévols.
BACKTRANSLATION	Mova el sis de trèboles al cinc de trèboles.
FINE TUNING	Moga el sis de trévols al cinc de trévols.
CA→ES	
REF	¿Quieres personalizar tu boda?
BACKTRANSLATION	¿Vamos a personalizar tu boda?
FINE TUNING	¿Quieres personalizar tu boda?
ES→PT	
REF	Remover o cinco de paus.
BACKTRANSLATION	Retirar o cinco de trébolas .
FINE TUNING	Remover o cinco de paus.
PT→ES	
REF	¿Puede un blog ser considerado literatura?
BACKTRANSLATION	¿Se puede considerar a un blogue literatura?
FINE TUNING	¿Puede un blog ser considerado literatura?
CA→PT	
REF	Cria os teus próprios detalhes de casamento
BACKTRANSLATION	Cria os seus próprios pormenores de casamento
FINE TUNING	Cria os teus próprios detalhes de casamento
PT→CA	
REF	No hi ha més jugades
BACKTRANSLATION	No existeixen més jogades
FINE TUNING	No hi ha més jugades

Table 14: Translation examples comparison between reference sentence, back-translation model and fine tuned model.

5 Lessons learned

In this section, we will discuss the results we have obtained from the Similar Language Translation task. We have to keep in mind that this study aimed to translate languages of the same family with low resources. These conditions made us use different techniques to improve the translation between all three languages.

Multilingual model compared to Bilingual model We have observed how the multilingual model has higher scores than the bilingual model. For the ES-CA directions, the improvement of the model is about +9 BLEU, which is a huge difference. For the PT-ES directions, we have an improvement of around +1 BLEU. For the CA-PT directions using pivot we see an improvement of around +3.5 BLEU.

Using the multilingual system, apart from the fact that thanks to the zero-shot, the system can do translations between Catalan and Portuguese and vice-versa without using the pivot technique; the model generalizes better and so the quality of the translations improves. The zero-shot makes the system achieve a BLEU score of 12 BLEU for the CA→PT and a score of 17 BLEU for the PT→CA direction.

Appreciating such improvement and being able to translate between Catalan and Portuguese we decided to use the multilingual model to make the improvements on.

Backtranslation improvement Having only parallel corpus data between ES-CA and ES-PT, we decided to use the backtranslation technique to take advantage of the monolingual data, by generating synthetic parallel data (pairing the sentences of the monolingual data generated with its translations).

As we show in Figure 7, we have an improvement of the multilingual model for all directions of between +0.5 to +3 BLEU, except when using CA monolingual data, which made worse the CA→ES direction score for -2 BLEU and the PT→CA direction for -0.03 BLUE in pivot case. This deterioration is probably due to the lower resemblance of the CaWaC dataset with the target domain. Which as we can see in Table 3, the TF-IDF score is the worse we have compared to all the other documents for the other languages.

When using this technique we have seen an improvement, but it takes a lot of time and resources to generate the translations from the monolingual data to all the language directions, and retrain the model. Nonetheless, it is a strong solution for Low-Resources languages, as we have seen for the translations between Catalan and Portuguese, that has improved +1 BLEU for the CA-PT direction and +2 BLEU for the PT→CA direction using zero-shot.

Domain adaptation improvement The last improvement we decided to add to our model has been the fine tuning domain adaptation. To do so, we added 2/3 of the domain set data to the already trained model, which has been the multilingual model with backtranslation. It was the best model we had so far. We have achieved an improvement for all directions of around +1 BLEU.

So using fine tuning helps to improve translation for an specific domain, and it can be done using less resources and less time than backtranslation does. Since we achieved a similar improvement using fine tuning compared to backtranslation, only by adding to the corpus of the model around 1.500 sentences. Whereas, when using backtranslation we needed to translate around 2.6 millions of sentences, which is a huge increment compared to the 1.500 sentences for fine tuning. After doing so for Backtranslation, we had to retrain all the model from scratch and for the fine tuning technique we only needed to retrain from the last checkpoint.

We would say that both techniques are great solutions for improving the translation for our models. But if the aim is to improve the model for an specific domain, and doing so as fast as possible and with minimum resources, then fine tuning is a great option, because it achieves a similar improvement compared to backtranslation, just by adding fewer data and spending less time training.

References

[1] Fifth conference on machine translation (WMT20). [Online]. Available: http://www.statmt.org/wmt20/

[2] Alex Sherstinsky. *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*, 2020. [Online]. Available: https://arxiv.org/pdf/1808.03314.pdf.

[3] Sepp Hochreiter and Jürgen Schmidhuber. *Long Short-Term memory*, 1996. [Online]. Available: https://www.bioinf.jku.at/publications/older/2604.pdf

[4] Felix Stahlberg, *Neural Machine Translation: A Review*, 2019. [Online]. Available: https://arxiv.org/pdf/1912.02047.pdf

[5] Jay Alammar, *The Illustrated Transformer*, 2018. [Online]. Available: http://jalammar.github.io/illustrated-transformer/

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, *Attention Is All You Need*. [Online]. Available: https://arxiv.org/pdf/1706.03762.pdf

[7] Fairseq. [Online]. Available: https://Fairseq.readthedocs.io/en/latest/

[8] Jonathan J. Webster, Chunyu Kit, *Tokenization as the Initial Phase in NLP* , 1992. [Online]. Available: https://www.aclweb.org/anthology/C92-4173.pdf

[9] Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, NandaKambhatla, *tRuEcasIng*, 2003. [Online]. Available: https://www.cs.cmu.edu/ llita/papers/lita.truecasing-acl2003.pdf

[10] Ivan Provilkov, Dmitrii Emelianenko, Elena Voita, *BPE-Dropout: Simple and Effective Subword Regularization*, 2020. [Online]. Available: https://arxiv.org/pdf/1910.13267.pdf

[11] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean, *Google's Multilingual Neural Machine Translation System: Enabling Zero-shot Translation*, 2017. [Online]. Available: https://arxiv.org/pdf/1611.04558.pdf

[12] Rico Sennrich, Barry Haddow, Alexandra Birch, *Improving Neural Machine Translation Models with Monolingual Data*, 2016. [Online]. Available: https://arxiv.org/pdf/1511.06709.pdf

[13] Chenhui Chu, Raj Dabre, Sadao Kurohashi, *An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation*, 2017. [Online]. Available: https://www.aclweb.org/anthology/P17-2061.pdf

[14] Bhargava Ram *Document-Similarity.* [Online]. Available: https://github.com/BhargavaRamM/Document-Similarity

[15] Diederik P. Kingma, Jimmy Ba *Adam: A Method for Stochastic Optimization.* [Online]. Available: https://arxiv.org/abs/1412.6980

[16] Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, Mikel Artetxe *Multilingual Machine Translation: Closing the Gap between Shared and Language-specific Encoder-Decoders.* [Online]. Available: https://arxiv.org/abs/2004.06575

[17] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu *BLEU: a Method for Automatic Evaluation of Machine Translation.* [Online]. Available: https://www.aclweb.org/anthology/P02-1040.pdf

[18] Magdalena Biesialska, Lluis Guardia, Marta R. Costa-jussà *The TALP-UPC System for the WMT Similar Language Task: Statistical vs Neural Machine Translation.* [Online]. Available: http://www.statmt.org/wmt19/pdf/54/WMT24.pdf

[19] Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, Yonghui Wu *Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges.* [Online]. Available: https://arxiv.org/pdf/1907.05019.pdf

A Multilingual Neural Machine Translation: the case-study for Catalan, Spanish and Portuguese Romance Languages

Multilingual Neural Machine Translation: Case-study for Catalan, Spanish and Portuguese Romance Languages

Pere Vergés Boncompte and Marta R. Costa-jussà
TALP Research Center
Universitat Politècnica de Catalunya, Barcelona
pere.verges@est.fib.upc.edu, marta.ruiz@upc.edu

Abstract

Machine translation is the task of automatically translating one language into another. We evaluated the translation between Catalan, Spanish, and Portuguese, which are Romance languages, using the Transformer architecture. We made use of different techniques to improve the translation between these languages. First of all, we used zero-shot to archive the translation between Catalan and Portuguese. Secondly, applied backtranslation to make use of the monolingual data. Lastly, to improve on the specific domain data, we have applied fine-tuning.

1 Introduction

Research in the field of Machine Translation (MT) is very active during the last years. From statistical approaches (Koehn et al., 2003) to neural ones (Bahdanau et al., 2015), the progress has been really impressive. Even if the latest neural architecture based only on attention mechanisms has achieved impressive results (Vaswani et al., 2017), there are still many challenges remaining, including multilingual translation from languages other than English and domain adaptation.

In these directions, the Similar Language Task organised in the context of Conference on Machine Translation (WMT 2020) provides a nice setting for these challenges. Within this task, the focus is translating between languages that are different from English and are language of the same family, including South-Slavic, Indo-Aryan and Romance.

In our case, we have devoted our research to the Romance languages, which include Spanish, Portuguese and Catalan. The evaluation comprised all translation directions, but only provided training data for Spanish-Portuguese and Spanish-Catalan. Therefore, we approached the Portuguese-Catalan pair from a pivot and zero-shot perspective.

2 Background

In this section we show an overview of neural-based multilingual machine translation and domain adaptation using fine tuning.

2.1 Multilingual translation

When having multiple languages, we can use different NMT architectures all based in the Transformer (Vaswani et al., 2017). Among the alternatives, we can share encoders and decoders (Johnson et al., 2017) or have specific encoders and decoders for each language (Escolano et al., 2020). In this paper, we are using the shared approach, while it would be left as future work to try other ones.

Shared encoder-decoder One direct approach is using the proposed universal encoder/decoder (Johnson et al., 2017) that lets us use Zero-Shot in a very simpler manner. This architecture is the same used for bilingual translation but we need to add a tag in the source sentence with the information of the target language, this tag has to be written in the following form (example for a translation from Catalan to Spanish):

```
<2es> Bon dia -> Buenos días
```

This is how parallel data has to be represented, we write a tag to indicate the target language, in the example would be Spanish. For the preparation we have all the source language sentences (with their corresponding tag) shuffled in one file and the parallel target data in another, and then all the data at once. Sharing all the parameters between all languages pairs makes the model generalize across languages which helps to improve the translation of the low resource languages pairs.

2.2 Monolingual corpus selection for backtranslation

There is a large amount of monolingual data available for the task. This monolingual data can im-

prove the system by using backtranslation (Sennrich et al., 2016). However, backtranslating takes time, so we decided to do data selection to use monolingual data most similar to the domain we wanted to translate. As a selection procedure, we used the TF-IDF (Term Frequency – Inverse Document Frequency), which determined the relevance of the words in a document. Using this information we have compared all the monolingual data that we were given against the development set, and keep the ones that had a higher score among all of them.

2.3 Domain adaptation

In order to have a better translation score on specific language domains, one approach is to do fine tuning of the model, that consists of retraining a model that was trained with out-of-domain data with in-domain data. One problem of fine tuning is that the in-domain data is usually small compared to the out-of-domain data and the model tends to overfit.
One way of trying to avoid the overfitting is to do mixed fine tuning, which consists on shuffling the in-domain with the out-of-domain data and training all the data together the same way you would do with a simple multilingual model (Chu and Dabre, 2019).

3 Experimental Framework

In this section we describe the data sets we used, the data preprocessing, the training and evaluation of the bilingual and multilingual systems.

3.1 Data and Preprocessing

All systems only use the data provided by the organizers, so we did not use any additional monolingual nor parallel data. For the Catalan-Spanish and Spanish-Portuguese translation we used all the parallel data available, which is about 11.3 million sentences for the Catalan-Spanish translation and 4.1 million sentences for the Spanish-Portuguese. For the Catalan-Portuguese we did not have any parallel data. We have also used monolingual data (documents names shown in Table 1), for the Catalan backtranslation about 2 million sentences, for the Portuguese about 1.1 million and for the Spanish about 1.5 million.

Preprocessing Our multilingual model is trained with all the parallel corpus data, and with pseudo-parallel data obtained doing backtranslation, we have translated all the monolingual data to the other languages and used it as the pseudo-parallel data, the translation was done by the best system at each moment (we did more than one backtranslation). To select which data to backtranslate, we have used TFIDF as measure of similarity [1]. This data was shuffled with the parallel one and we did a split of 1 to 100 of training and validation of all the data. Then we did the standard procedure for preparing the data, which is normalizing, tokenizing, and trained and applied truecasing to the data using the *Moses*[2] scripts. After applying these scripts we have also used another script from *Moses* which is the *clean-corpus-n.perl*, which cleans the corpus, we have limited the length of the sentences from 1 to 50 and the maximum length difference from one sentence to the other to be 1.5. Then for preparing the data we have learned the byte-pair encoding (BPE)[3] for the concatenation of the source and target languages. Vocabulary is shared among all languages.

Monolingual corpus for backtranslation
News-commentary-v15 ES
News-crawl-2015 ES
News-commentary-v15 PT
News-crawl-2019 PT
CaWaC CA

Table 1: Documents used for the backtranslation

3.2 Parameter Details

Bilingual and Shared Multilingual Our models are based on the Transformer architecture implemented from the *fariseq* toolkit [4], we have used specifcally the *transformer_iwslt_de_en* architecture, which is also implemented in the toolkit.
The following parameter where used for the configuration of the model: six attention layers for the encoder and the decoder, with four attention heads per layer with an embedding dimension of 512 and sharing the decoder input output embedding, for the multilingual model we also share all the embeddings. Each batch has a maximum number of tokens set to 2048. We used the Adam optimizer setting the betas to $\beta_1 = 0.9$ and $\beta_2 = 0.98$, with a learning rate of 0.0005 varied with the inverse square root of the step number, and a warm-up

[1] https://github.com/BhargavaRamM/Document-Similarity
[2] https://github.com/moses-smt/mosesdecode
[3] https://github.com/rsennrich/subword-nmt
[4] https://github.com/pytorch/fairseq

steps equal to 4000. With a dropout of 0.1 and a weight decay and gradient clipping norm to 0, we have used as criterion the label smoothed cross entropy set to 0.1

4 Results

We have compute the scores using 1/3 of the development set made available in the task. We exclude 2/3 of the development set to use it for fine-tuning. We show the progress of the results we have obtained applying the different techniques to improve the translations of our final system. In order to be able to compare the improvement between translation between Catalan-Portuguese (CA-PT), we decided to show the scores of theses directions using the pivot technique, for each direction what we did was translate CA->ES->PT (to get the CA->PT translation) and PT->ES->CA (to get the PT->CA translation).

First of all we compared the performance of the bilingual system against the multilingual. This comparison is shown in Table 2. The score archived by the multilingual model using zero-shot for CA->PT is 12.47 and for PT->CA is 17.67 BLEU. In

Directions	Bilingual	Multilingual
ES-CA	64.23	73.12
CA-ES	60.64	69.56
ES-PT	27.20	27.62
PT-ES	29.7	30.57
CA-PT	20.99	24.94
PT-CA	25.21	28

Table 2: BLEU results of bilingual and multilingual systems.

the Table 2, we see how the multilingual model performs better than the bilingual. We picked the multilingual model as the best one we had so far and applied backtranslation using the monolingual data mentioned in Table 1. In Table 3 we show the scores achieved by the multilingual vs the multilingual with backtransaltion. The backtranslation model using zero-shot has achieved for the CA->PT direction a score of 19.64 and 13.56 BLEU for the PT->CA direction. We achieved a for all directions except for the ES->CA a better BLEU score using backtranslation. So we applied fine tuning to the backtranslation technique. In table 4, we show the results of backtranslation model vs backtranslation with fine tuning model. The score achieved using zero-shot for CA->PT direction is 16.05 and for the CA->PT 19.64 BLEU.

Directions	Multilingual	Backtranslation
ES-CA	73.12	70.59
CA-ES	69.56	73.01
ES-PT	27.62	28.80
PT-ES	30.57	30.89
CA-PT	24.94	25.52
PT-CA	28.00	27.97

Table 3: BLEU results of multilingual and multilingual with backtranslation systems.

Directions	Multilingual	Backtranslation
ES-CA	70.59	71.21
CA-ES	73.01	74.05
ES-PT	28.80	29.55
PT-ES	30.89	32.12
CA-PT	25.52	26.94
PT-CA	27.97	29.18

Table 4: BLEU results of multilingual with backtranslation and multilingual with backtranslation and fine tuning systems.

5 Discussion

We now will discuss the results obtained for each system we have trained.

Multilingual model compared to Bilingual model We have observed how the multilingual model has higher scores than the bilingual model. For the ES-CA directions, the improvement of the model is about +9 BLEU, which is a huge difference, and for the PT-ES directions, we have an improvement of around +1 BLEU. For the CA-PT directions using pivot we see an improvement of around +3.5 BLEU.

Using the multilingual system, apart from the fact that, thanks to the zero-shot the system can do translations between Catalan and Portuguese and vice-versa without using the pivot technique; the model generalizes better and so the quality of the translations improves. The zero-shot makes the system achieve a BLEU score of 12 BLEU for the CA->PT and a score of 17 BLEU for the PT->CA direction.

Backtranslation improvement Having only parallel Corpus between ES-CA and ES-PT we decided to use the backtranslation technique to make use of the monolingual data. Making synthetic parallel data pairing the sentences of the monolingual data generated with its translations.

As we show in Table 3, we have an improvement of the multilingual model for all directions of between +0.5 to +3 BLEU, except when using

CA monolingual data, which made worse the CA->ES score for -2 BLEU and the PT->CA direction for -0.03 BLEU in pivot case. This deterioration is probably due to the lower resemblance of the CaWaC dataset to the target domain, which we have calculated using TF-IDF score.

Domain adaptation improvement Lastly, we have applied fine tuning in order to do domain adaptation. To do so we added 2/3 of the development set data to already trained model, which has been the multilingual model with backtranslation since it was the best model we had so far. We have achieved an improvement for all directions of around +1 BLEU.

So using fine tuning helps improving translation for an specific domain, and it can be done using less resources and less time than backtranslation does. Since we achieved similar improvements using fine tuning compared to backtranslation, only adding to the corpus of the model around 1500 sentences whereas in order to use backtranslation we needed to translate around 2.6 millions of sentences. After doing so for backtranslation we had to retrain all the model from scratch and for the fine tuning technique instead of retrain the model from scratch, it is only need to retrain from the last checkpoint. Both techniques have been great solutions for improving the translation for our models, but fine-tuning has been more effective in terms of resources and time.

Acknowledgments

We are grateful to Carlos Escolano for his comments, corrections and help throughout the investigations. This work is supported in part by the Catalan Agency for Management of University and Research Grants (AGAUR) through the FI PhD Scholarship. This work is supported in part by the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund, the Agencia Estatal de Investigación through the postdoctoral senior grant Ramón y Cajal and the projects EUR2019-103819, PCIN-2017-079 and PID2019-107579RB-I00.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Chenhui Chu and Raj Dabre. 2019. Multilingual multi-domain adaptation approaches for neural machine translation. *CoRR*, abs/1906.07978.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2020. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. abs/2004.06575.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of the Conference of the NAACL*, pages 48–54.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

B GEP

Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona (FIB)

Final degree project

GEP

Author:
Pere Vergés Boncompte

Supervisor:
Marta Ruiz Costa-Jussà

June 4, 2020

Contents

1 Introduction

This project aims to have a reference of how Transformer perform with machine translation when they are used to translate languages that come from the same origin. Transformer is a model that uses attention to increase the speed of the training of this kind of models, attention what does is allows the model to look for words that are near the one that it is actually processing, this way the word that is treating has a context and references to other words, and this attention helps the model to make better encoding for each word, since it has the information of the context it is in, so the Transformer is similar to and encoding-decoding model but in each encoder and decoder it has a self-attention layer that helps to achieve this better word encoding [17].

In this project all three languages are romance. These languages are Spanish, Catalan, and Portuguese, we already have a model that translates from Spanish to Catalan and from Spanish to Portuguese, so in this project, we would like to achieve the translation between these three, especially the machine translation between Catalan and Portuguese.

The first part will reproduce the results we already have in the translation of Spanish and Portuguese, using Transformer with help from fairseq, a python library that allows us to train easily machine translation models and that already has the Transformer architecture implemented. Then the second part would be to do the same between Spanish and Catalan.

The last one, which is what we could say the principal aim of this project, is to use the zero-shot translation to achieve the translation between all three languages. The idea behind zero-shot is to classify data in classes that are disjoint from the training instances. The zero-shot learning has two steps, the first one is training, in our case the training step is training the models of Catalan-Spanish and Spanish-Portuguese, the second step is doing the inference, in our case is extracting the knowledge of both models to generate the translation between Catalan and Portuguese.

The results of this project will be a reference on how good is Machine Translation using Transformer on similar languages, this will help the scientific community on this filed, to have an overview of how this models perform.

2 Context

This task is proposed by the EMNLP 2020, it is the fifth conference on machine learning (WMT20), the one I will be focusing on is the Similar Language Translation Task. Last year's task was similar to this one, since there was Similar Language Translation between Spanish and Portuguese, with a score performance of around 0.60 for both directions of translation Spanish to Portuguese and Portuguese to Spanish [19][20]. Last years task was only on the translation between Spanish and Portuguese, this year task apart from this task we also have to translate between Spanish and Catalan and then do the translation using zero-shot learning, so we are not going to adapt the solution of last year because we want to see how Transformer performs in this kind of task with similar languages.

2.1 Basic concepts: Natural language processing

Natural language processing is the study and process of large amounts of natural language data. This can be used to translate automatically one language to another, to analyze the sentiment of a sentence, to predict answers to questions, to generate natural language... [1][2].

There are different types of tasks in NLP. The following are some examples:

2.1.1 Language modelling

Language modeling is the task of calculating the probability of a sequence of words and can be used to calculate the probability of the next word in a sequence or sentence. To do this we have that we need to represent languages in some way, we can represent it by sentences, words, letters... this fragmentation of the text varies depending on which task you want to apply it, but this is called tokenizing and apart from fragmentation the text it also normalizes it removing dots, cases... To calculate the probability for the next word it uses the probability chain rule, but using the Markov Assumption that says that you only need one or two words as a prefix to reduce computations.[10][11]

2.1.2 Machine Translation (MT)

Machine translation is the task of automatically translating one natural language to another while preserving the meaning and intention of the text, and it is a Language Modelling that is conditioned to a source sentence. There are different models of Machine Translation one of the most famous one is seq2seq modelling, this models consists of two RNN that one is an encoder and the other one the decoder, this model works entering a sequence of words (a sentence) and it outputs another sequence of words, that is the translation of the first sequence. But these models usually do not perform well when treating long sentences, that is why we are using Transformer, because with the help of the self-attention

layers we encode the words with relations with words in the same sentence, making a better encoding with more information of the context [3][4].

BLEU evaluation There are different systems of evaluation for machine translation systems, such as the Word Error Rate, which calculates the number of words that differ from a reference translation or LEPOR. But We will be using the BLEU metric to evaluate our model, the BLEU metric correlates highly with human evaluation, and the idea behind it is to evaluate a sentence comparing it to a professional human translation, so the closer to this translation it is the higher the score is [29][30].

BLEU is very similar to precision, in machine translation the precision is between ngrams, but it does some modifications to be able to compare it to multiple references

$$P = \frac{common\, ngrams}{reference\, ngrams}$$

The modification of the precision is done by adding a penalty for length difference between translation, because otherwise a sentence that has two words that are very common such as The, as, be, and a long sentence that contains such words would have a precision of 1[31]. This modification works as such:

$$PB = \begin{cases} 1 & \text{if c} > \text{n} \\ e^1 - \frac{r}{c} & \text{Otherwise} \end{cases}$$

So the formula to calculate the BLEU score is the following:

$$BLEU = PB \times exp(\sum_{n=1}^{N} w_i \, logP_n)$$

Tokenization In machine translation, you have to encode words in some way. To do this usually, words are transformed into tokens. There are several ways of tokenizing a word, but the approach that fairseq usually uses and what we are going to use is BPE (Byte Pair Encoding), which is a data compression algorithm that compresses pairs of common characters and replaces them by a single character that does not appear in the data, for example, you could substitute the pair "ou" for the single letter "A" if "A" did not appear in your data, this way we can compress the data and train it in a faster way. Using subword-NMT that uses BPE is a very good approach with languages that have the same alphabet, this way the subword segmentation is consistent in all of the languages involved in the translation [32][33].

2.1.3 Deep Learning techniques

To solve this problem, the approach I have to take is using Deep Learning. In this field, there are two main types of neural networks that are often used to

tackle these problems. Recurrent-Neural-Network and a sub-type of this kind of networks that are the Long-Short-Term-Memory RNN.

Recurrent-Neural-Network (RNN) In RNN the input is usually a word, which differs from a standard neural network, this helps the network to acquire more flexibility of the length of the sentence input, not only this but also shares the features learned in different positions of the text input.

There are three types of structures of RNN and are: many-to-one, one-to-many and many-to-many, the names are quite self-explanatory. [12][13]

Long-Short-Term-Memory (LSTM) These are a special kind of RNN because they solve the problem of RNN, since in each cell they regulate the information they want to remember and which one they want to forget, and can have long term memory.

In contrast to a standard RNN, an LSTM have four layers in each cell. This cell structure allows the neural network, to forget some past information (this step is done by the forget gate layer), and then has another two gates which decide which information the cell is going to store.[12][13]

2.2 Stakeholders

This project's aim is not to build or sell a product, this project consists of research to help the NLP community have more data about Machine Translation between languages of the same origin, in this case, romance. So basically the target of this research is all the NLP community interested in this issue. So we could say that the users of this project are other researchers to have a reference and even use the system, but there will not be a graphical interface or anything similar to use the results of this research. The researchers will benefit from knowing if the approach that I will take is better than other ones or not and the reasons why it performs the way it will do.

2.3 State of the art

State of the art results for Machine Translation nowadays can be achieved using an architecture of Transformer[14], which is an encoder-decoder architecture not is only that but you also have self-attention layers, as we have already said helps to make better encoding of the words because is creates relations between words and adds this information in the encoding of the word.

For my task, I have also to take into account the use of zero-shot to achieve a translation between languages which we do not have data, as we explained earlier zero-shot has two steps, the first one is learning from a supervised data-set in our case we have data from Spanish to Catalan and from Spanish to Portuguese, so we have to extract all the knowledge from this data-sets into our models and

the make inference of this data to achieve a translation where we do not have data, in this case from Catalan to Portuguese. There are no previous studies in this matter so I will be the first to try the translation between Catalan and Portuguese using the zero-shot between Catalan-Spanish and Spanish-Portuguese using Transformer.

3 Scope

The main target of this study is to apply new multilingual techniques in the translation of similar languages that can be used for zero-shot translation. The idea behind zero-shot translation is to train with known attributes and the do inference to a new set. What we want to do is use the zero-shot translation to take advantage of the knowledge we have on translation from Spanish to Portuguese and from Portuguese to Spanish and from Spanish to Catalan and from Catalan to Spanish and with this knowledge do inference and learn the translation from Catalan to Portuguese and from Portuguese to Catalan. [21][22]

To test the performance of this project I will use the test set that I will be given by the challenge itself and see how it performs, to train the neural network I will use the Transformer architecture since it is the best one there is right now.

To make everything more clear I will now list all the tasks that I will intend to perform:

- **Task 1**: Translation between Spanish and Portuguese, this first task I will be using fairseq to make the model for the machine translation, the architecture as mentioned earlier will be using Transformer. The results are intended to be similar to the ones that were achieved in last year's competition.
- **Task 2**: Translation between Spanish and Catalan, this second task I will be using fairseq as well, to make the model for the machine translation, the architecture is also going to be using Transformer. The results are intended similar to the ones we achieve in the first task.
- **Task 3**: Now that we have the models that translate from Catalan to Spanish and from Spanish to Portuguese we want to make use of the zero-shot technique to achieve a translation between Catalan and Portuguese, in this case, we have no reference on how this performance will be, but this is the aim of the project, to see how Transformer work with similar languages in a Machine Translation task.

3.1 Obstacles

First of all, we can take into account obstacles, like having an illness or some health problem while working in this project, in this case, I would have to stop the project and retake it when these problems are solved. Another could be that my laptop or persona computer broke, but this problems is easily solvable, I could buy another one or use the computers that we have in the university.

But the main issues that we are facing are is the zero-shot translation task, which means that we do not have training data between Portuguese and Catalan, we do have data between Spanish and Portuguese and between Spanish

and Catalan. So we are going to provide translation utilizing multilingual techniques. Another problem is that the zero-shot technique does not work properly or its performance is very low, then we should look to have another approach to this problem and try different techniques, this could mean a lot of extra time, to study which technique could work, learn it and then apply it to our problem.

Other obstacles that we can face are that the neural network takes to much time than I have expected so some tasks may need to be rescheduled, or that the Calcula cluster has some kind of problem and I could not use it. This means that I do not have enough computational power to do all the necessary computations for this project, this could be solved by renting to google cloud some clusters to do the computations but this probably will not be necessary.

There could also be that the fairseq library had some bug and I had to work around it to evaluate the model correctly, and another issue that can arise with libraries is that there is not enough documentation or bad documentation and I had to spend a lot of time trying to understand how to use it.

3.2 Methodology and rigor

The methodology that I will approach is agile doing all the projects step by step and adapt in case the circumstances of the project change. In my case dividing this research into steps means breaking it into steps of translation, since the zero-shot approach needs the models of the machine translation between Catalan-Spanish and Spanish-Portuguese.

The first steps will be to train the neural network of this to translation working properly with a good performance, and when this is done do the final step that is to merge these two neural networks and build the translation between Catalan-Portuguese-Spanish. This one is the most difficult part. For each part the idea is to document it at the end of each step, this way we will have better documentation since it will be written after we have done the training and checked the results with my tutor. After the last task, I will have to check that everything is right and try to make all the documentation coherent.

As explained earlier to validate that each model translates properly, first we have the BLUE score, which gives us a good insight on how well does the system perform, this score is between 0 and 100 so having a score above 50 is quite good, apart from this score, we can also use the model and see if the translation make sense, we can do this using fairseq with the command fairseq-interactive, if you provide fairseq with a model and the BPE codes, then you can use the model to do translations from one language to the other one.

3.3 Tools

The tools that I will be using are fairseq, that is a sequence modeling written in PyTorch that in our case we will use it for language modeling and translation and supervised Transformer, that has an encoder and a decoder but the important part is that has a self-attention layer that boosts its performance. These are libraries that are implemented in python3, this is an open-source programming language. Apart from these tools, I will also need a text editor, to program. For the documentation I will be using Latex, in my case, I will be using a web tool, which is overleaf, this way I can access my archives whenever I want. To monitor all the work I will be doing I will use Github, a version control tool that will allow me to save easily all my versions of my code and access them easily.

For hardware tools, I will use the Calcula clusters, this way I can access GPU to train the neural network, and my laptop to program, here we could also consider hardware some papers and pens, and of course a printer and we could also consider all the electricity to run the clusters, laptop, and printer.[14][15][16]

4 Planning

4.1 Estimated duration of the project

This project is estimated to be performed in about four months since it starts at the beginning of February and ends around the end of June, in case I needed more time to do the whole project I could submit it in October so I could have four more months to do it.

4.2 Resources

Here I will explain all the resources I will need to perform this project. Since this is a research the main resource I will need, is time.

4.2.1 Software resources

Now I will state all the programs and software resources that I will need and use to perform this project.

- Ubuntu 18.04 and Unix editors (such as vim, nano...) and programs as well as Windows.
- Python 3, Pytorch, Fairseq, which are programming languages and libraries.
- Latex and overleaf, this is going to be my text editor for the documentation, then I will also use vi and nano to program and write scripts.

4.2.2 Hardware resources

This project does not require a lot of hardware resources.

- Calcula cluster from UPC to train the deep neural network.
- My own laptop to do all the scripts and documentation, which is a Xiaomi Air 12', and my desktop pc.
- Office material, such as paper sheets or pens, and of course a printer in order to print the documentation.
- We could also take into account the electricity we use for running the cluster, laptops and printer.

4.2.3 Human resources

For this project, the only human resources that are required are myself, to train and develop all the projects, and the help of my project manager, the supervision of the GEP professor. We could also take into account the designer of the problem and the data scientist that collected all the data to train the models.

4.3 Tasks

As I stated before I will use and agile strategy so now I will state all the tasks that I will have to take in this project.

4.3.1 Learning

Before starting this project I had little knowledge of deep learning and none from NLP, so I need to learn about deep learning and NLP, not only as a theory but also the programming structure and how to treat data. From the programming part, there is a need to learn how pytorch and fairseq work, which parameters they need and how do they need the data for the input to train the model.

4.3.2 Starting the project

The first steps of the development of this project are familiarizing with the cluster and interpreting the results that fairseq generates from this model. To do it, we are going to try an example from the fairseq documentation in Calcula and see how everything works, how the data needs to be preprocessed, and how the network learns.

4.3.3 GEP

The five first weeks of this course will be necessary to do the GEP course, which will help organize all the project, this is an important step since it will give an overall image of how much time the project will take, which steps are needed to take and plan them as accurately as possible, not only this but also helps as a start of the documentation. Basically, the resources needed to do this are a laptop and any text editor to do the documentation.

4.3.4 First task

The first task to develop is starting to translate from Spanish to Portuguese and vice-versa since we already have information of other models and its performance, the idea is to match the same level of accuracy. All the data to do this task is gathered from last year's challenge, and not from this year since it has not been released. Once this first task is met we can start the second task. For this one, the resources will be my laptop to write the scripts and the Clacula cluster. This task starts around the 24th of February and is expected to be finished for the 23rd of March, but it may be left the network training after this period ends to achieve even better results.

4.3.5 Second task

The second task consists of starting to translate from Spanish to Catalan and vice-versa, this second task is quite similar to the last one and should take around the same time, but here we met a problem that the data may not be released so it may be hard to train a model without the data of the challenge. If

there is no data from the challenge we have to options, one is looking for other data, or make and upgrade of the architecture of the first task. This tasks apart from the training of the network, we have all the documentation work which takes a lot of time as well.

4.3.6 Third task

This is the difficult task which will take the biggest load of work in this project, here we will need to use the zero-shot to make the network learn from the models we have trained before and learn from the translation of Spanish-Portuguese and Spanish-Catalan we need it to learn the translation of Spanish-Portuguese-Catalan. If the second task takes the time expected which to be finished for the 6th of April, it would take around a month or more so for around the middle of May this third should be finished.

4.3.7 Final task

Here we would need to polish everything and put it right if it took less time of the one that I have expected I could take on another task and try to solve it as well. But it depends of who everything is dealt with. This one would take from the start of May to its end since the networks will probably have better results than they had before or not, but either way, I will have to check it and interpret the results and explain them in the documentation, the documentation will take a lot of time since I have to test the code and be precise when interpreting the outputs of the model.

4.3.8 Time approximations

Now I will show how much time I think each task is going to take me, I may be wrong and that some tasks take more time than I have expected or less. These are the ours that I will spend on each task, but the neural network will for sure spend more time training.

Task	Hours
GEP	75 hours
Learning	25 hours
1st task	75 hours
2nd task	75 hours
3rd task	125 hours
Final task	50 hours
Delivery	25 hours
Total	450 hours

The only tasks that can be done concurrently are the GEP and learning, I could also do the first and second task at the same time, but since the challenge has not already uploaded the data I can not start with the second task. But I think

this is the best way because I can try different approaches for the first task and choose the best one to apply it to the second task. After the second task, all the other ones are dependent on its predecessors, to do the zero-shot approach I first need to have the models of the first and second task finished and working with a minimum performance.

4.3.9 Alternatives

There is the possibility that some obstacles come up, as we stated before, all the problems that we have stated can be solved the same way, spending more time with this project. So if we are behind schedule I will have to spend more time than I had expected to this project, if this delay is little, probably adding around to 100 hours are probably sufficient to make up for the delay. For alternatives on how to approach the project are little since we want to see how Transformer perform in similar languages, if I do not manage to make it work I can not work around it. The only approach that I could change if I run into trouble is the zero-shot, then I would have to look for another technique.

If I run into a lot of problems I would have to postpone it and deliver the project in October, so I could spend another 400 hours in this project and try to solve everything with this extra time.

4.3.10 Gannt chart

This project has a value of 18 ECTS, from which 3 are of the GEP, so what we have is that an ECTS credit is equivalent to 25 hours, so if we can take this as an indication of how much hours we should spend doing the final project. We have that the GEP will take 75 hours and the rest of the project will take 375 hours. The distribution of the tasks is represented in the following figure (Figure 3), which is the Gannt chart. I approximately spend 2 to 3 hours per day on this project.

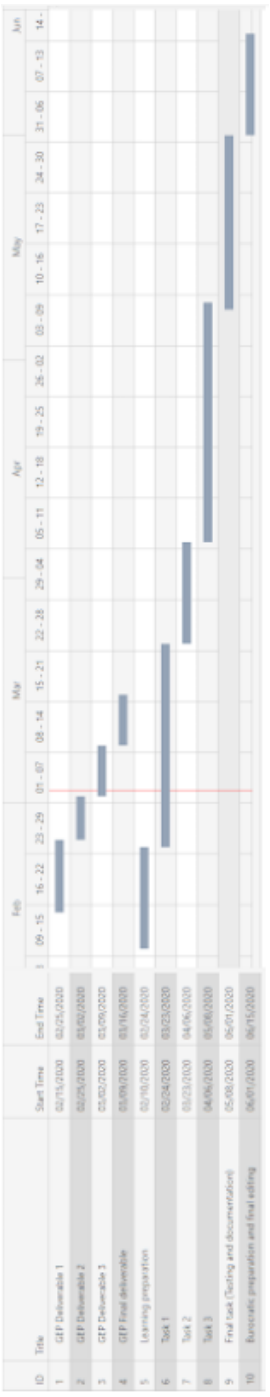

Figure 1: Gannt chart.

5 Economic management

To estimate the budget of this project we are going to take into account both the direct and the indirect costs. These will be linked with all the tasks that we have stated before.

5.1 Direct costs

In this project, we can divide the direct costs in basically three main parts, like we did when we mentioned the resources needed for this research, which are software, hardware, and human resources.

5.1.1 Software costs

The programs that I used to develop this project are mainly open source. On my laptop I have Ubuntu 18.04, which is open source and free, then on my other computer I have Windows 10 home which has a cost of 120€, so for the operating systems, the cost is 120€.

Then the programs and libraries used for this project are python, fairseq, then we have Overleaf and google drive that also has a free version, the same goes for IDEs. Last but not least we have all the datasets that are also open source and free.

So for what software expenses we only have the expense of buying Windows 10 license, which could be avoided by using Windows 10 without a license.

$$Software\,costs = 120€$$

5.1.2 Hardware costs

The hardware that I have used for this project is basically my laptop, my desktop computer and we could also say the energy they have consumed. I bought my laptop in June of 2018, and my computer in December of 2018.

These costs can be amortized following the following formula:

$$\frac{Price}{Lifespan \times Working\,days\,per\,year \times Working\,hours\,per\,day} \times Hours\,used\,in\,this\,project \tag{1}$$

- Laptop

$$\frac{700€}{5\,years \times 220\,days/year \times 8\,hours/days} \times 225\,hours = 22.37€$$

- Desktop computer

$$\frac{1050€}{7\,years \times 220\,days/year \times 8\,hours/days} \times 225\,hours = 19.17€$$

The final total cost of the usage of my computers is:

$$22.37 + 19.17 = 41.54€$$

This cost is always constant in the project since all the part of the project are done using a computer either my laptop or my desktop computer, I decided to divide equally the number of hours that I will use my laptop and my desktop computer.

The other hardware that is needed to deal with this project is the usage of the Cluster, this is considered a service since we do not buy the cluster itself but it is usually rented to a company. So to figure out the cost of the usage of the cluster we will approximate by the cost it would have if we rent it to google cloud, which is $0.41 per GPU per hour since I have access to 2 GPU we have to double the cost, which is $0.82/hour and in euros is €0,74/hour. In this case, the hours of usage of the cluster will be quite larger than the project hours since the cluster will still be running and training the network while we are not working on it, my approximation is that the use of the cluster will be around 500 hours for all the tasks.

$$0.74€/hour \times 500\,hours = 370€$$

$$Hardware\,costs = 411.54€$$

5.1.3 Human costs

This cost will be calculated by the average hourly price of the role needed for each task in Spain. In this project we could say that we have three main roles, the first and most common in any project is the Project Manager, then we have the Data Scientist, this is an important role since treating right the data is crucial, and the last one is a software developer.

- **Data scientist**: In Spain, the average salary is of 35.800€ per year, if we approximate this value to an hourly rate it becomes to 18€ per hour worked. [21]
- **Project Manager**: In Spain, the average salary is of 41.200€ per year, if we approximate this value to an hourly rate it becomes to 22€ per hour worked.[22]
- **Software engineer**: In Spain, the average salary is of 32.800€ per year, if we approximate this value to an hourly rate it becomes to 16€ per hour worked.[23]

Now that we know how much every worker on average earns we can divide the tasks and assign them a number of hours to deal with each task, the following table show the distribution of tasks for each team member.

Task	Hours	Data Scientist	Project manager	Software eng
GEP	75 hours	15 hours	60 hours	0 hours
Learning	25 hours	20 hours	5 hours	0 hours
1st task	75 hours	45 hours	5 hours	25 hours
2nd task	75 hours	45 hours	5 hours	25 hours
3rd task	125 hours	80 hours	5 hours	40 hours
Final task	50 hours	40 hours	5 hours	5 hours
Delivery	25 hours	5 hours	15 hours	5 hours
Total	450 hours	250	100 hours	100 hours

We have calculated approximately how many hours each member of the team will spend doing this project, now we can calculate the cost of having such a team, the following table show this calculations:

Position	Hourly rate	Hours worked	Total cost
Data scientist	18€/hour	250 hours	4500€
Project Manager	22€/hour	100 hours	2200€
Software engineer	16€/hour	100 hours	1600€
Total	-	450 hours	8300€

We have to apply the SS cost which will increase the cost of human cost about 30%, if we take that into account we have the following cost:

$$Human\,costs = 8300 \times 1.3 = 10790€$$

The total cost of the direct cost is the sum of software, hardware, human costs, which is:

$$Total\,costs = 120 + 411.54 + 10.790 = 11321.54€$$

5.2 Indirect costs

The indirect costs that we can consider are basically the electricity cost, the Internet connection, and some office supplies like paper sheets, pens and the cost of printing documents.

The electricity consumption will be considered from the usage of my computers only, which my laptop consumes around 40W and my desktop computer 650W. Since before we partitioned the consumption time of both personal computer as equal we can calculate the average of both and apply all the calculations on

the average. Then we could calculate the electricity consumption of the cluster but since we have calculated its price as a service it is included already on the direct cost of the cluster. For the internet connection we have that in Spain the average cost of an internet connection is around 35 euros per month, so we will use this number to approximate the cost of the internet connection[24]. The last part is the office supplies, this it is a cost that I will approximate as a total since most of the months the cost will be little, but the last one will be quite huge since I will have to print the whole documentation of the project and make copies of it and bind all of them, and this is quite expensive, an average price for three copies of a 100 pages bind is about 100 euros, so I would say that the whole cost of the office supplies will be around 150 euros [25].

$$Average\,computers\,consumption = \frac{40W + 650W}{2} = 345W$$

$$Average\,Spain\,electricity\,cost\,0.2874€/kWh$$

$$Electricity\,usage\,450\,hours$$

$$Electricity\,total\,cost\ = 450hours \times 0.345kW \times 0.2874€/kWh = 44.61€$$

$$Internet\,total\,cost\ = 4months \times 35€/month = 140€$$

$$Office\,supplies\,total\,cost = 150€$$

The total cost of the indirect cost is the sum of electricity, internet, office supplies, which is:

$$Total\,costs = 44.6 + 140 + 150 = 334.6€$$

5.3 Cost contingency

Most of the obstacles that can arise in this research can be resolved by spending more time doing the task, so the main issue would be the human resources for this project since the software is not likely to fail and if it did we could look for another free available software (but it is very unlikely to fail), and if the hardware fails, for example, the Calcula cluster dies, we could then rent it to google cloud and the cost would be the same. Even the cost would not change very much I will now calculate the possible extra cost that could be involved if there is some kind of contingency:

- We do not have enough computational power and we need more GPUs to do the training of different networks at the same time, then we could rent google cloud more cluster, this cost would be 0.37 euros per hour. So the additional cost of this problem would be 0.37 per the hours needed to do the extra computations.

- Calcula cluster dies, if this happens we would rent GPU to google cloud and the cost would be exactly the same.
- If my laptop dies, then I could only use my desktop computer and the electricity cost would be almost double. I can not calculate an exact number because it depends on when the laptop would die if it was my desktop computer the one that dies then the electricity cost would decrease.
- If I am behind schedule and I have to spend more time in the project this would probably be around 100 hours. If this was the case then we would have a lot of extra costs:
  - Human resources:

$$Extra\,costs = 100 \times 18.6€/hour \times 1.3 = 2418€$$

  - Clusters: This would be the same.
  - Computer amortization:

$$Extra\,costs = 3.98 + 4.26 = 8.24€$$

  - Electricity:

$$Extra\,cost\ = 100hours \times 0.345kW \times 0.2874€/kWh = 9.91€$$

  Total extra cost of 2436.15€.
- If the 100 extra hours are not enough to deal with the problem then I would have to deliver the project next semester which would cost the same amount of doing the project without contingencies. So there would be an extra cost of 9166.14€.

The following table explains the risk exposure of the cost contingency on this project:

Risk	Probability	Extra cost	Risk cost
Behind schedule (100h)	60%	2436.15€	1561.07€
Behind schedule (450h)	30%	11659.14€	3497.74€
Total	-	-	4959.43€

5.4 Total costs

The following table show the total cost of this project:

Item	Cost
Direct cost	11321.54€
Indirect cost	334.6€
Contingency cost	4959.43€
Total	16615.03€

5.5 Control management

Basically in this project, the main cost is the human resources so the best way of balancing the cost of this project is monitoring the number of hours that are actually spent. In order to do this, I will keep track of how many hours I spend each day on the project, and at the end of each task, I will check if the number of hours is more or less than I have expected. This calculation is simple as subtracting the expected hours to the real number of hours spent in the task and multiplying it by the average hourly cost.

6 Sustainability report

The sustainability matrix has three main parts which are project put into production, exploitation, and risks, for each of this blocks we have an environmental, economic and social impact.

	PPP	Exploitation	Risks
Environmental	Consumption of design: 1	Ecological footprint: 5	Environmental: -5
Economic	Invoice: 5	Viability plan: 15	Economic risk: -4
Social	Personal impact: 9	Social impact: 10	Social risk: 0
Sustainability rank	15	30	-9

6.1 Auto enquiry

When developing an IT project, usually I do not think about the environmental consequences that it can represent, but that is because usually IT project are only using resources such as a laptop and a server, these obviously are not good for the environment since they break and end up in containers, both server hardware or a personal computer, and they are not easy to recycle. But in the case of a computer even if I do not make an IT project I would have a laptop and it would represent the same environmental danger. For what I think about social repercussions usually, these projects try to make people's lives easier so the idea is to help people and not harm them, this is one of the main things that are in my mind when developing a project. Making applications and project equal for everybody usually involve a cost that for an individual or a little company can not pay, so I probably do not think of everybody when doing a project, but this is not because I want to discriminate somebody but because I do not have the resources and time and remuneration to do so. So usually when developing an app or project I think about the average user and maybe my projects are not suitable or usable for a minority. I would say that taking into account all the environmental, social and equality aspects in a project is actually very hard to do and to estimate the harms of each aspect not only because it is very time-consuming and expensive but also because it is also unpredictable in some cases.

6.2 Economic dimension

The approximation of the project, I think that I have kept all the costs fairly realistic. I think that the cost will be quite accurate, the only things that may not be realistic in that a data scientist or a project manager would want to actually take part in this project and work part-time. But apart from this issue, the budget that we have calculated is quite realistic.

Regarding the lifespan of our research, I do not think that it will be very useful in 5 or 10 years since there will be upgrades in the Transformer and there will be better models for training, nonetheless, it will be a good reference for

checking it these new models are actually better than the old ones, and how do they perform when training with languages that evolved from the same source language

Economically my solution is free since it is a nonprofit project, but it will be better than other ones since it will require less data and less training iterations to do a good performance, so it will use fewer resources such as clusters and electricity.

6.3 Environmental dimension

As I have already said when commenting on the budget, we have a laptop and a desktop computer which will be working for around 450 hours in total, which has a consumption of electricity. Then we have the cluster if we have that an Nvidia GPU consumes about 0.35kWh and in this case, we will use it for 500 hours or so we will consume using the cluster 175kW [27]. Then we have all the paper and ink used in the final delivery of the project that will be about 300 pages and ink for these 300 pages. So there is not a big environmental impact involved in this project.

6.4 Social dimension

The benefits for society in this project are achieving a model of machine translation between Catalan, Spanish and Portuguese achieved using Transformer. This helps because it will be a reference for how well Transformer perform with similar languages. Apart from having a model that can translate between these languages.

Apart from these benefits for the scientific society as a reference for the performance of Transformer models, it will also be beneficial to me since I will have learned how this system works, how neural networks work and finishing my first big project.

References

[1] "Natural language processing" On: Wikipedia [Online]. [21 of February 2020]:
https://en.wikipedia.org/wiki/NaturalLanguageProcessing

[2] "What is Natural Language Processing?" On: WhatIs? [Online]. [21 of February 2020]:
https://searchbusinessanalytics.techtarget.com/definition/natural-language-processing-NLP

[3] "Machine Translation" On: The Stanford NLP Group [Online]. [21 of February 2020]:
https://nlp.stanford.edu/projects/mt.shtml

[4] "A Must-Read NLP Tutorial on Neural Machine Translation – The Technique Powering Google Translate" On: analyticsvidhya [Online]. [21 of February 2020]:
https://www.analyticsvidhya.com/blog/2019/01/neural-machine-translation-keras/

[5] "Natural language inference" On: NLP-progress [Online]. [21 of February 2020]:
http://nlpprogress.com/english/naturaLlanguageInference.html

[6] "Natural Language Inference and NLP" On: Hackernoon [Online]. [21 of February 2020]:
https://hackernoon.com/natural-language/inference-and-nlp-xq47230md

[7] "Coreference resolution" On: NLP-progress [Online]. [21 of February 2020]:
http://nlpprogress.com/english/coreference_resolution.html

[8] "State-of-the-art neural coreference resolution for chatbots" On: Medium [Online]. [21 of February 2020]:
https://medium.com/huggingface/state-of-the-art-neural/coreference-resolution-for-chatbots-3302365dcf30

[9] "Coreference Resolution" On: The Stanford NLP Group [Online]. [21 of February 2020]:
https://nlp.stanford.edu/projects/coref.shtml

[10] "Learning NLP Language Models with Real Data" On: Medium [Online]. [21 of February 2020]:
https://towardsdatascience.com/learning-nlp-language/models-with-real-data-cdff04c51c25

[11] "NLP: Explaining Neural Language Modeling" On: Github [Online]. [21 of February 2020]:

https://mchromiak.github.io/articles/2017/Nov/30/Explaining/Neural-Language-Modeling/XlFunuko85k.

[12] "NLP: Explaining Neural Language Modeling" On: Github [Online]. [21 of February 2020]:
https://towardsdatascience.com/natural-language/processing-from-basics-to-using-rnn-and-lstm-ef6779e4ae66

[13] "Understanding LSTM Networks" On: Github [Online]. [22 of February 2020]:
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[14] "The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)" On: Github [Online]. [23 of February 2020]:
http://jalammar.github.io/illustrated-bert/

[15] "fairseq" On: fairseq [Online]. [23 of February 2020]:
https://fairseq.readthedocs.io/en/latest/

[16] "Fairseq" On: Github [Online]. [23 of February 2020]:
https://github.com/pytorch/fairseq

[17] "The Illustrated Transformer" On: Github [Online]. [12 of March 2020]:
http://jalammar.github.io/illustrated-transformer//

[18] "NLP: Explaining Neural Language Modelingr" On: Github [Online]. [12 of March 2020]:
https://mchromiak.github.io/articles/2017/Nov/30/Explaining-Neural-Language-Modeling/#.XmoEx6hKiUk//

[19] EMNLP 2020 FIFTH CONFERENCE ON MACHINE TRANSLATION (WMT20) On: statmt [Online]. [25 of February 2020]:
http://www.statmt.org/wmt20/

[20] ACL 2019 FIFTH CONFERENCE ON MACHINE TRANSLATION (WMT19) On: statmt [Online]. [25 of February 2020]:
http://www.statmt.org/wmt19/similar.html

[21] What Is Zero-Shot Learning? On: aim [Online]. [25 of February 2020]:
https://analyticsindiamag.com/what-is-zero-shot-learning/

[22] Average Data Scientist Salary in Spain On: PayScale [Online]. [5 of March 2020]:
https://www.payscale.com/research/ES/Job=Data_Scientist/Salary/9b2d8f8e/Barcelona

[23] Average Project Manager, (Unspecified Type / General) Salary in Spain On: PayScale [Online]. [5 of March 2020]:
https://www.payscale.com/research/ES/Job=Project_Manager\%2C_(Unspecified_Type_\%2F_General)/Salary/a0107bba/Madrid/

[24] Average Software Engineer Salary in Spain On: PayScale [Online]. [5 of March 2020]:
https://www.payscale.com/research/ES/Job=Software_Engineer/Salary

[25] Internet access in Spain On: JustLanded [Online]. [5 of March 2020]:
https://www.justlanded.com/english/Spain/Spain-Guide/Telephone-Internet/Internet-access-in-Spain/

[26] Encuadernacion con Espirales On: pixartprinting [Online]. [5 of March 2020]:
https://www.pixartprinting.es/impresion-revistas-catalogos-libros/espiral-metalico/

[27] Electricity Prices in Spain On: Electricity in Spain [Online]. [5 of March 2020]:
https://electricityinspain.com/electricity-prices-in-spain/

[28] POWER REQUIREMENTS FOR GRAPHICS CARDS On: RealHardTech [Online]. [5 of March 2020]:
https://www.realhardtechx.com/index_archivos/Page362.htm

[29] Evaluation of machine translation On: Wikipedia [Online]. [9 of March 2020]:
https://en.wikipedia.org/wiki/Evaluation_of_machine_translation#Automatic_evaluation

[30] BLEU: a Method for Automatic Evaluation of Machine Translation On: Aclweb [Online]. [9 of March 2020]:
https://www.aclweb.org/anthology/P02-1040.pdf

[31] BLEU: On: Wikipedia [Online]. [9 of March 2020]:
https://es.wikipedia.org/wiki/BLEU

[32] Subword NMT On: Github [Online]. [9 of March 2020]:
https://github.com/rsennrich/subword-nmt

[33] Byte Pair Encoding — The Dark Horse of Modern NLP On: Github [Online]. [9 of March 2020]:
https://towardsdatascience.com/byte-pair/encoding-the-dark-horse-of-modern-nlp-eb36c7df4f10