BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

BILINGUAL DICTIONARY BASED NEURAL MACHINE TRANSLATION

NEURÁLNÍ STROJOVÝ PŘEKLAD ZALOŽENÝ NA BILINGVÁLNÍCH SLOVNÍCÍCH

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **MAKSIM TIKHONOV**
AUTOR PRÁCE

SUPERVISOR **SANTOSH KESIRAJU, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

146753

Institut:	Department of Computer Graphics and Multimedia (UPGM)
Student:	**Tikhonov Maksim**
Programme:	Information Technology
Specialization:	Information Technology
Title:	**Bilingual Dictionary Based Neural Machine Translation**
Category:	Speech and Natural Language Processing
Academic year:	2022/23

Assignment:

1. Get familiar (theory & implementation) with sequence-to-sequence architectures for neural machine translation.
2. Pick one of the standard datasets and obtain baseline supervised MT results.
3. On the same dataset, obtain results using only biligual dictionary based NMT (baseline system, eg: from XLM).
4. Implement the anchored training approach (see the reference literature for the methods).
5. Compare the methods and analyse the results.
6. Propose and implement an extension, if time permits.

Literature:

1. Duan et al, "Bilingual Dictionary Based Neural Machine Translation without Using Parallel Sentences". ACL 2020.
2. Conneau and Lample, "Cross-lingual Language Model Pretraining". NeurIPS 2019.
3. Wang et al, "Expanding Pretrained Models to Thousands More Languages via Lexicon-based Adaptation". ACL 2022

Requirements for the semestral defence:

- Points 1-3

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Kesiraju Santosh, Ph.D.**
Head of Department:	Černocký Jan, prof. Dr. Ing.
Beginning of work:	1.11.2022
Submission deadline:	10.5.2023
Approval date:	31.10.2022

Abstract

The development in the recent few years in the field of machine translation showed us that modern neural machine translation systems are capable of providing results of outstanding quality. However, in order to obtain such a system, one requires an abundant amount of parallel training data, which is not available for most languages. One of the ways to improve the quality of machine translation of low-resource languages is data augmentation. This work investigates the task of Bilingual dictionary-based neural machine translation (BDBNMT), the basis of which is the use of the augmentation technique that allows the generation of noised data based on bilingual dictionaries. My aim was to explore the capabilities of BDBNMT systems on different language pairs and under different initial conditions and then compare the obtained results with those of traditional neural machine translation systems.

Abstrakt

Vývoj v oblasti strojového překladu v posledních několika letech ukázal, že moderní neuronové systémy strojového překladu jsou schopny poskytovat výsledky vynikající kvality. Pro získání takového systému je však zapotřebí velké množství paralelních trénovacích dat, která nejsou pro většinu jazyků k dispozici. Jedním ze způsobů zlepšení kvality strojového překladu pro low-resource jazyky je augmentace dat. Tato práce zkoumá úlohu neuronového strojového překladu založeného na bilingválních slovnících, jejíž základem je použití augmentační techniky umožňující generování zašuměných dat na základě bilingválních slovníků. Mým cílem bylo prozkoumat možnosti systémů založených na této metodě na různých jazykových párech a za různých výchozích podmínek a následně porovnat získané výsledky s výsledky tradičních neuronových systémů strojového překladu.

Keywords

Artificial intelligence, natural language processing, machine translation, neural machine translation, bilingual dictionaries, bilingual dictionary based neural machine translation, low-resource machine translation, training

Klíčová slova

Umělá inteligence, zpracování přirozeného jazyka, strojový překlad, neurální strojový překlad, bilingvální slovníky, neurální strojový překlad založený na bilingválních slovnících, low-resource strojový překlad, trénování

Reference

TIKHONOV, Maksim. *Bilingual Dictionary Based Neural Machine Translation.* Brno, 2023. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Santosh Kesiraju, Ph.D.

Rozšířený abstrakt

Moderní neuronové modely strojového překladu jsou schopny poskytovat výsledky srovnatelné s výsledky profesionálních lidských překladatelů. K vytvoření takového modelu je však zapotřebí obrovské množství paralelních dat (desítky milionů paralelních vět), aby na nich mohl být model natrénován. Pro většinu jazykových párů však takové množství paralelních dat neexistuje. Jedna z metod pro zlepšení výkonnosti modelů, které nesplňují tyto podmínky je použití augmentačních technik.

Tato práce zkoumá použitelnost augmentačních metod založených na použití bilingválních slovníků v kontextu neuronového strojového překladu (NMT) pro různé scénáře.

Jednou z metod využití bilingválního slovníku, která je pokrytá v této práci je generování pseudoparalelních dat na základě slovníku. Tato práce se zabývá analyzou následujících případů použití této metody za předpokladu omezeného množství paralelních dat (130 tisíc vět) a různého množství dodatkových monolignválních dat: trénování MT modelu od nuly na malém množství dodatkových dat, trénování MT modelu od nuly na velkém množství dodatkových dat a adaptace veřejně dostupného jazykového modelu (XLM-R) na malém množství dodatkových dat.

Jinou zkoumanou metodou je trénování pomocí kombinací *Anchored training* (AT) a *Anchored Cross-lingual Training* (ACP). Tyto metody byly představeny v článku Duan et al. [1] a tamtéž byly prozkoumány pro případ přítomnosti rozsáhlých monolingválních korpusů. Tato práce zkoumá přínosnost trénování pomocí zmíněných metod na malém množství neparalelních dat.

Další průzkum, který byl v této práci proveden, se týká použitelnosti předtrénování modelu s výše zmíněnou kombinací metod ACP a AT s následnou adaptací pro úlohou MT na paralelních datech. Výsledky tohoto způsobu trénování jsou pak porovnány s tradičními metodami předtrénování, které předpokládají jazykové modelování jako předtréninkový krok.

Bilingual Dictionary Based Neural Machine Translation

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. Santosh Kesiraju, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Maksim Tikhonov
May 9, 2023

Acknowledgements

I would like to thank my supervisor Santosh Kesiraju, Ph.D. for his advice and patience throughout this year.
I also want to thank the Speech@FIT research group and prof. Dr. Ing. Jan "Honza" Černocký personally for providing the access to computing cluster for the purpose of conducting experiments.
Finally, I would like to thank my parents for their financial and moral support.

Contents

Chapter 1

Introduction

Since the beginning of humanity's written history people were separated by the so-called curse of Babel. People of different cultures and ethnics group speaking different languages were incapable of communicating with each other on a somewhat decent level without a translator. With the invention of computing machines came the idea of delegating this work, or at least part of it, to machines. So Machine translation was born.

The quality of translation produced by MT systems was substantially improving over the last decades. Firstly, with the introduction of Statistical machine translation systems [2] which, unlike the preceding rule-based approach, were able to generate translations based on statistical models whose parameters are derived from the analysis of bilingual text corpora. The second improvement came with the introduction of end-to-end neural encoder-decoder MT systems in 2013, which marked the dawn of neural machine translation (NMT).

Modern-day NMT systems show a translation quality that is comparable to that provided by professional translators. But, to achieve such results NMT system should be trained on huge parallel corpora (also known as bitexts). Not meeting this condition results in subpar performance, comparable to, or even inferior to, the performance of SMT systems.

Data augmentation is one of the most commonly employed methods for enhancing the performance of MT systems. Augmentation techniques such as Back-translation [3], and synthetic data generation have been shown to effectively increase the amount and diversity of training data, leading to improvements in translation quality. In addition to data augmentation, other strategies such as the fine-tuning of pre-trained language models, and incorporating domain-specific knowledge [4] have also been explored to enhance the performance of MT systems.

The novel NMT approach – Bilingual dictionary based neural machine translation (BDBNMT) [1] – utilizes a synthetic data generation technique that leverages the bilingual dictionary. This approach considers the absence of parallel corpora, while it is possible to utilize large-scale monolingual corpora and ground-truth bilingual dictionary to close the gap between two languages by establishing the anchoring points via using the mappings provided by the dictionary.

This work investigates the capabilities of the following bilingual dictionary based techniques: *Anchored Cross-lingual pretraining* (ACP) and *Anchored training* (AT) described in [1], and the training of language model on pseudo-parallel data generated by the substitutions provided by a bilingual dictionary.

The main goal of this work was to examine the capabilities of these methods to provide performance improvement for models for different language pairs under various initial conditions:

i. Training the models from scratch with various amounts of additional monolingual data.

ii. Training the models from scratch with a limited amount of monolingual data.

iii. Fine-tuning the pre-trained *XLM 17* language model with a limited amount of both parallel and non-parallel data.

Chapter 2

Machine translation background

This chapter focuses on providing the reader with information concerning machine translation, its history of development, recent Neural machine translation developments and the translation quality evaluation metrics that are usually utilized.

2.1 Pioneers of machine translation

Machine translation is a task of natural language processing that involves the automatic translation of text from one language to another using computer algorithms. The history of machine translation is traced from the early systems of the 1950s when the first experiments were carried out. One of these experiments, the Georgetown–IBM experiment, despite the facts that this experiment was of demonstrative nature and the product of it had almost no real practical application (the system had only 6 grammar rules and a very restricted vocabulary of 250 words to translate sentences in Russian that were prepared in advance into English), the demonstration had great success. This, along with excessively optimistic estimations that automatic translation would be a solved problem within a decade, resulted in a significant increase in funding and garnered substantial attention from the press and scientific community.

However, in the next few years, these first-generation MT systems had very limited success. They consisted of a large bilingual dictionary where the entry for each word in the source language was provided with equivalents in other languages and syntactic rules that were used to place the output words in the right order. After years of research, it became evident that systems of this kind are not capable of doing a high-quality translation. Mainly because the solution was not systematic: syntactic rules and a bilingual dictionary just weren't enough to make the system solve the problem consistently in various domains because of languages' semantic ambiguity and overly complex rules.

In 1960 Yehoshua Bar-Hillel, a linguist and a pioneer in the field of MT, argued that MT systems back then were not capable of fully automatic high-quality translation [5]. He stated that in different kinds of documents, there could be sentences "whose ambiguity is resolvable only on the basis of extra-linguistic knowledge", those sentences, in opposite to scientific documents or reports, are more prone to misinterpretation by MT systems without aforementioned knowledge. His statements turned out to be right when six years after the Automatic Language Processing Advisory Committee (ALPAC), following the examination of the current state and prospects of MT, concluded in its report that the products of research in the MT field are disappointing and that "there is no immediate or predictable

prospect of useful machine translation". This report justified a reduction in government funding for MT research and it was abandoned for years to come.

2.2 Years of quiet

Despite the impact that the ALPAC report had in the US, active research in the MT field continued in Europe and Canada to satisfy the needs of local governments and companies. For the next 20 years, there was a development of the Rule-based machine translation (RBMT) approach. RBMT systems are based on linguistic information of source and target languages provided by dictionaries and grammars covering syntactic, morphological and semantic aspects of languages. Although there was progress in translation quality, coverage of different languages and introduction of new techniques – Transfer based machine translation, Interlingual machine translation – systems still had crucial shortcomings: to create one, a whole staff of linguists was required in order to build new rules and dictionaries. Also, they still were lacking the ability to generalize across different domains and were mainly geared up towards the translation of scientific papers and reports.

Some of the many successful systems of the time:

- **SYSTRAN** Russian-to-English MT system made for the United States Air Force, was also used as assistance in the translation of scientific documents written in Russian.
- **METEO** MT system designed for the translation of weather forecasts between English and French languages issued by the Department of environmental policies of Canada.

2.3 Statistical machine translation

The introduction of the groundbreaking Statistical Machine Translation (SMT) approach in 1990 [2] marked a significant turning point, leading to rapid changes in the MT field. The proliferation of the Internet and the accumulation of parallel corpora, i.e. collections of aligned sentences in two languages (also known as bitext), in previous years, created conditions that enabled MT systems to move beyond predefined rules and bilingual dictionaries but instead to be created in a process of training on large parallel corpora. It was at this point that the approach to MT began to leverage machine learning methods.

In practice, this meant that although the objective of translating a sequence of tokens (sentence) from a source language L_{src} with vocabulary $V_{L_{src}}$ to the most semantically similar sequence of tokens in the target language L_{tgt} with vocabulary $V_{L_{tgt}}$ remained the same, the method for achieving it shifted from the application of rules that constitute an MT model to a different approach. Now, instead of the manual creation of these rules, they are extracted from large parallel corpora via training.

SMT model's translation of a sequence from the source language to the target one is determined by the estimation of the probability that some sequence in the target language is an accurate translation of this source sequence. The search for the best translation is based on Bayes' rule:

$$\underset{y}{argmax}\ P(y|x) = \underset{y}{argmax}\ \frac{P(x|y)P(y)}{P(x)} = \underset{y}{argmax}\ P(x|y)P(y), \tag{2.1}$$

where

$P(y|x)$ is a probability, that sequence y is a translation of sequence x.

$P(x|y)$ is a the probability of how often we see a x given that y is seen. It is defined by the translation model.

$P(x)$ and $P(y)$ are the probabilities of some sentences in source/target languages defined by language models.

The parameters of these models are estimated during the training. Model training itself can be described as a process of improving the performance of a model for a specific task (such as MT) by leveraging the training data. In the case of SMT, one of the most widely used training procedures was the Minimum error rate training [6].

2.4 Neural machine translation

The Neural machine translation (NMT) approach was explored and evaluated back in the 1990s [7, 8] and despite the fact that the approach was very similar to the methods that we use today, its results were unsatisfactory and neural method was considered subpar and ineffective compared to SMT, that was dominant in those years. From today's perspective, fail of early NMT systems seems predestined: none of these models was trained on corpora of size large enough.

The modern incarnation of NMT started taking effect in 2007 with the integration of neural language models into SMT systems [9] and, after some time, the major breakthrough for MT came with the introduction of pure NMT with a sequence-to-sequence model in 2014 [10, 11] which, after some additions – attention mechanism [12] and byte pair encodings [13], was able to outperform its SMT competitors at shared task for MT at WMT16. NMT continued its development and the most recent architectural development emerged in 2017 with the introduction of the Transformer in the "Attention is all you need" paper by Vaswani et al. [14].

2.4.1 Natural language representation

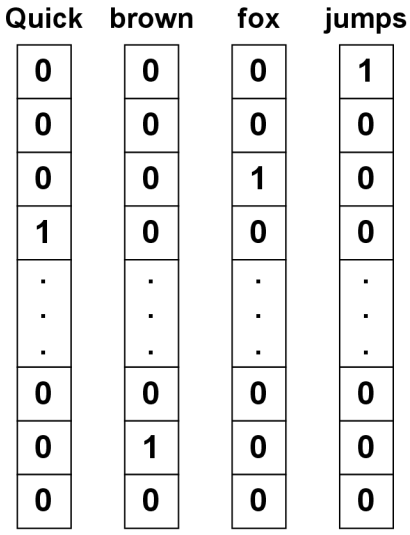
Before we move on to NMT architectures, I would like to review the modern techniques that are being used to facilitate the machine's understanding of natural language.

Byte pair encoding Before we feed the input text to the model, it should be tokenized. One of the most efficient and most popular methods (used in BERT [15], GPT [16] and many other systems) to tokenize the input text is byte pair encoding (BPE) [17]. With this subword-based method, we obtain the vocabulary which consists of tokens during the training. A token can either be a full word or a subword from a training corpus. To indicate the end of the word, the algorithm adds the special token (e.g. "*</w>*") to the end of the token.

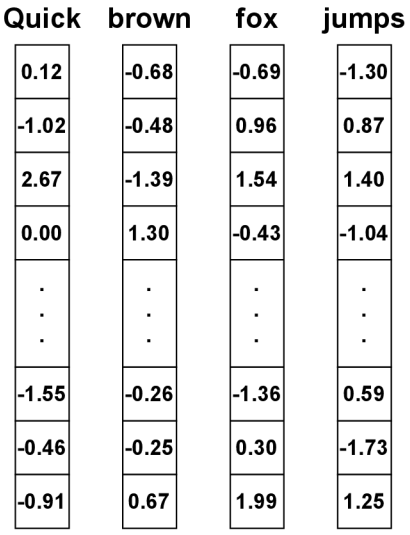
Having some limitations regarding vocabulary size, the BPE algorithm ensures that only the most frequent words and subwords extracted from rare words are present in the vocabulary. For example, the morphologically complex German word *Sonnensystem* ("Solar system") may be split into separate subwords *Sonn*, *en* and *system</w>* which can be used for the tokenization of other words that are compound of at least one of these subwords.

This is a far better approach than, for example, word-based tokenization because it allows us to reduce the size of the vocabulary and the number of OOV (out-of-vocabulary) word encounters in test data. Moreover, the nature of the BPE algorithm, that it tries to catch the most frequent patterns from the training corpus, implies that it will capture the semantically relevant language units (prefixes, suffixes, root words, etc.).

Word Embeddings *Word embeddings* is a fundamental concept in NMT. They are continuous vector representations of words in a high-dimensional space, where each dimension of a vector corresponds to a specific feature or attribute of the word. The main idea behind word embeddings is to capture the semantic and syntactic relationships between words in a way that natural language processing systems (NMT systems in particular) can easily process.



(a) One-hot word vector representations. The dimensionality of word vectors is equal to the size of the vocabulary. Vectors are extremely sparse, each has a value of 1 at the index of the corresponding word and 0 at other positions. Vectors bear no semantic information.



(b) Word embedding vectors. This type of word representation offers us a drastic reduction in word vector dimensionality and a contextual similarity between word vectors.

Figure 2.1: Illustration of different word representations.

Since semantics and syntax are concepts that are hardly understandable by the computer, embedding algorithms adopt the approach that is best described by the famous quote of J.R. Firth [18]:

> *You shall know a word by the company it keeps.*

And thus, to install the aforementioned word dependencies, we define the meaning of a word by the context in which it occurs. Words that occur in similar contexts (e.g. words "fox" and "wolf") will have similar embedding representations.

2.4.2 Sequence-to-sequence models

At a high level, the sequence-to-sequence (Seq2Seq) model is an end-to-end model which consists of two components: an encoder and a decoder. Both components traditionally utilise the RNN architecture (being stacked layers of either long short-term memory cells (LSTM) or gated recurrent unit cells (GRU)) and, unlike all non-neural translation systems before, are trained jointly, so they can learn the same context vectors to maximize the translation performance.

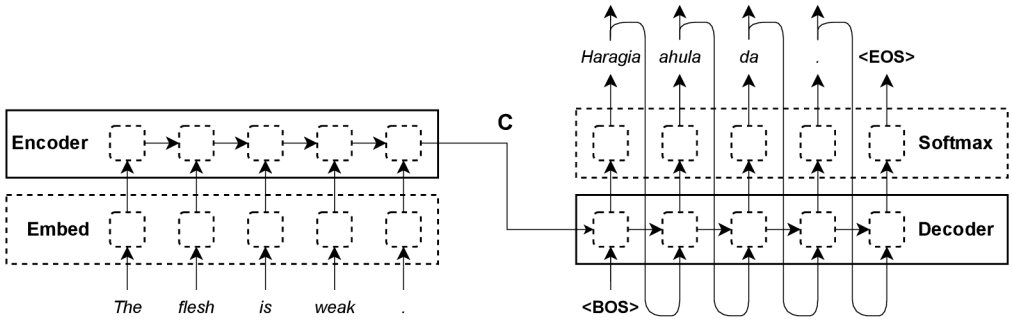


Figure 2.2: Example of Seq2Seq model translating from English to Basque. Illustration is inspired by [19].

Encoder The purpose of an encoder is to read the variable-length input sentence and to produce a fixed-length representation called the *context vector*. To do so, the encoder first of all maps words in a sentence to vectors of ones and zeros (one-hot representation). Then, using the embedding layer, the encoder maps each one-hot encoded input token to a dense vector representation – embedding. Once the input tokens are transformed into dense vector representations, the context vector $\mathbf{c}$ can be computed step by step for each input token using the deep RNN (LSTM or GRU models)

$$\mathbf{h}_i = f(\mathbf{W}_{hx}\mathbf{x}_i + \mathbf{W}_{hh}\mathbf{h}_{i-1} + \mathbf{b}_h), \tag{2.2}$$

where

$\mathbf{h}_i$ is the hidden state of RNN at timestep i. The hidden state of the last RNN layer is the context vector.

$\mathbf{x}_i$ is the input embedding vector at timestep i.

$\mathbf{W}_{hx}$ is a learnt weight matrix, that integrates an input vector.

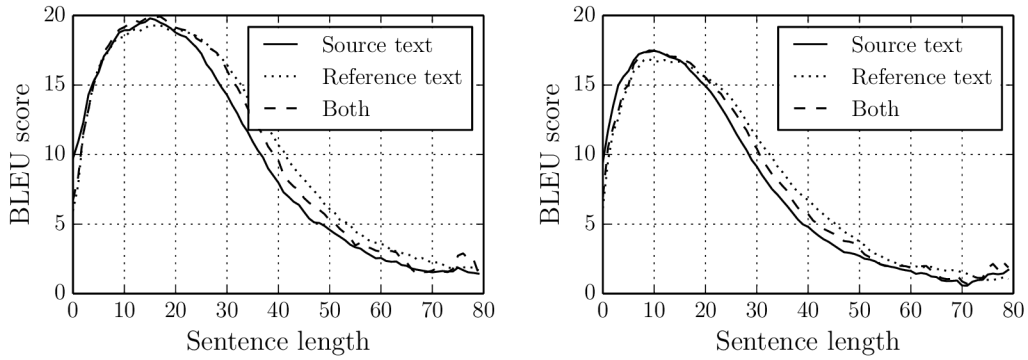
$\mathbf{W}_{hh}$ is a learnt weight matrix, that integrates a vector from the previous timestep.

$\mathbf{b}_h$ is a bias term.

Decoder Decoder, having the same architecture as an encoder, has a different task: it is responsible for generating the output sentence based on the context vector provided by an encoder. Its first layer's hidden state is being initialised with context vector $\mathbf{c}$ provided by the encoder. Then decoder at each timestep, given the context vector $\mathbf{c}$ and all the previously predicted words (at the first step decoder only has the <BOS> token), runs all layers of LSTM/GRU and applies the softmax function after that to generate the next

word. If a newly generated word is <EOS> (end of sequence) token generation ends. It is important to note that due to the way that output is being generated there is no relation between the length of the input sequence and the length of the output sequence.

Revealed problems Although the new Seq2Seq approach has shown promising results, after analysing the NMT systems' performance, Cho et al. [20] discovered several drawbacks. They found that, while being on par with traditional SMT systems under favouring conditions, the Seq2Seq systems' performance degrades rapidly: first, with the increase of the number of unknown words, and second, with the increased length of source sentences (see Figure 2.3).



(a) RNNenc (b) grConv

Figure 2.3: The BLEU scores achieved by the RNN Encoder–Decoder [11] (a) and by the Gated recursive convolutional neural network (b) depending on the length of the sentence obtained in the study of [20]. Degradation of performance with sentence length increase is evident.

And while the first problem may be potentially solved by increasing the size of vocabularies used by NMT systems, the second one, caused by the fact that the fixed-length vector representation does not have enough capacity to encode a long sentence with complicated structure and meaning, required the redesign of the system's architecture itself.

2.4.3 Sequence-to-Sequence models with attention mechanism

To address the described problem of deteriorating performance for longer sentences, [12] proposed an extension to the encoder-decoder model which learns to align and translate jointly.

Encoder Encoder, unlike the one in the vanilla encoder-decoder architecture, now utilises a bidirectional recurrent neural network (BiRNN), which consists of *forward* and *backward* RNNs: the former one reads the input sequence as it is ordered (from x_1 to x_N) and computes a sequence of *forward* hidden states $(\overrightarrow{h_1}, ..., \overrightarrow{h_N})$ for every input word, the latter one does the same computation except it reads the input sequence in reverse order and thus provides the *backward* hidden states $(\overleftarrow{h_1}, ..., \overleftarrow{h_N})$. Then we obtain *word representation* h_i for each word x_i by concatenating i-th forward and backward hidden states $h_i = \begin{bmatrix} \overrightarrow{h_i} \\ \overleftarrow{h_i} \end{bmatrix}$.

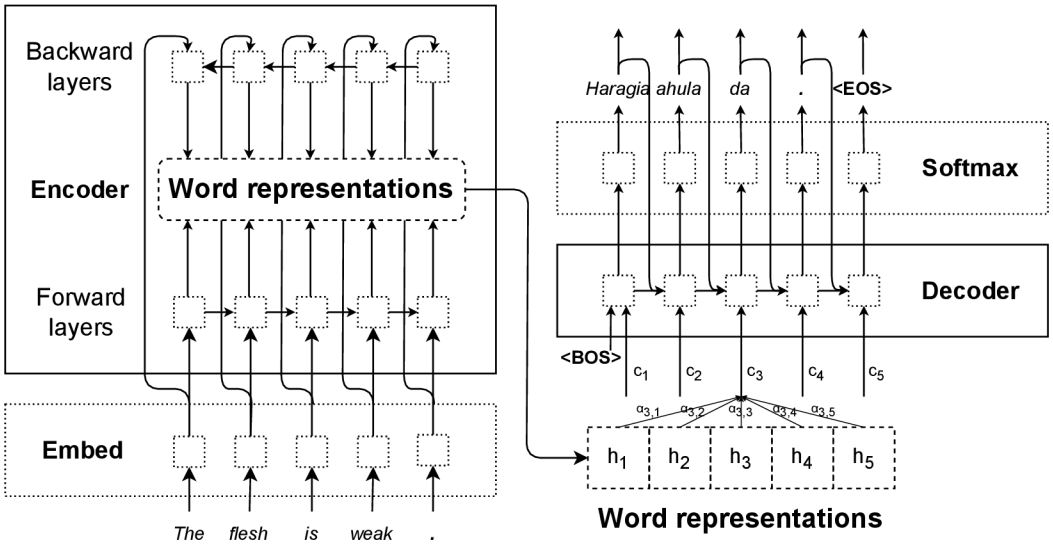


Figure 2.4: Illustration of RNN encoder-decoder with the attention mechanism.

This way each word representation h_i will contain information about both the preceding and the following words with a strong emphasis on the i-th word. And thereby encoder is now able to capture dependencies between the i-th token and the tokens that are from both sides of it.

Decoder In the proposed architecture a decoder is almost identical to the one from the original Seq2Seq model's architecture, except now for each step i, in addition to the previously predicted target words (y_1, ..., y_{i-1}) and the previous hidden state s_{i-1}, it should also take in account the *attention*. That is, the probability of the target word is:

$$p(y_i|y_1, ..., y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i) \quad \text{with} \quad s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{2.3}$$

The attention has the same role as context vectors and is also denoted as c_i for each step i. But now, instead of providing a single representation for the entire sentence for every step of the output generation, we, having a single vector of *word representations*, are evaluating it differently depending on the current target position i with the help of distinct weights. In essence, the attention is a weighted sum of word representations $\mathbf{h}$:

$$c_i = \sum_{j=1}^{N} \alpha_{ij} h_j \tag{2.4}$$

where α_{ij} is a specific weight that represents the probability that target word y_i is aligned with source word x_j. It is a product of *softmax* function over all other alignments:

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{N} \exp e_{ik}} \quad \text{with} \quad e_{ij} = a(s_{i-1}, h_j) \tag{2.5}$$

where e_{ij} is an alignment model which estimates how well the inputs around the j-th position match the i-th output word.

This approach removed the need to encode the entire source sentence into a single context vector by allowing the decoder to look through all the encoder states and evaluate

the importance of each source word for a current step. As a result, the model achieved superior performance when translating longer sentences, thereby solving the problem that motivated the creation of extension in the first place, and improving the overall translation quality establishing the Seq2Seq approach (and thus NMT as a whole) as the dominant paradigm.

2.4.4 Transformer

The next breakthrough in NMT came along with the introduction of a novel model architecture in the famous "Attention Is All You Need" paper from Google Research in 2017 [14]. This architecture, the Transformer, in order to overcome the constraint of sequential computation was designed purely around the attention mechanism which is used to extract dependencies between input and output. This new method permits parallel computing, which drastically speeds up the model's training since GPUs that are used for its training are specifically designed for parallel processing.

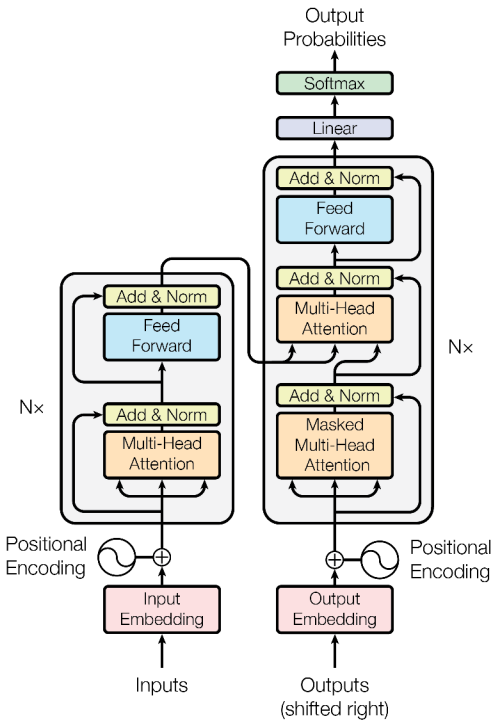


Figure 2.5: The Transformer model architecture. Taken from the original paper [14].

Transformer's architecture follows the general idea of encoder-decoder structure: it consists of two same "macro" blocks – an encoder and a decoder, the former one maps the input words $(x_1, ..., x_n)$ to their continuous representations $(z_1, ..., z_n)$. The latter one generates an output sequence $(y_1, ..., y_m)$ based on these representations and previously predicted words. Both of these blocks consist of N ($N = 6$ in the original paper) encoder/decoder layers.

Encoder layers Each encoder's layer consists of two sublayers – *multi-head self-attention mechanism* and *position-wise fully connected feed-forward neural network* (FFNN). The first

one relates different positions of a sentence to compute a representation of it. The second one applies non-linear transformations to its input features.

In addition to this, encoder layers also include *residual connections* around both sublayers followed by *layer normalization*. Residual connections allow the model to learn more complex functions by adding the input to the layer's output, and layer normalization helps stabilize the training process. That is, each sublayer has the output

$$\mathbf{y} = LayerNorm(\mathbf{x} + Sublayer(\mathbf{x})) \tag{2.6}$$

where $SubLayer()$ is the sublayer's function.

Decoder layers Decoder layers are structured in the same way as the encoder layers – they also utilize residual connections and the normalization step after each sublayer. However, the decoder layers also contain another sublayer that is placed between the two sublayers that are included in the encoder layer – the *Encoder-Decoder attention* sublayer – which performs multi-head attention over the output of the encoder stack. Additionally, the self-attention sublayer is masked which restricts the decoder from attending to the positions that were not yet generated during the training.

Attention The Transformer's attention can be described as a function with 3 arguments, all of them being vectors: query $\mathbf{q}$, key $\mathbf{k}$ and value $\mathbf{v}$ of dimensions d_k, d_k, d_v, respectively; the output of this function is computed as a weighted sum of values $\mathbf{v}$, where the weight assigned to each element in it is computed by a compatibility function of the query with the corresponding key.

The attention used in the original Transformer is called "Scaled Dot-Product Attention", it is computed with the above-described query, key and value as arguments, with the difference that in practice these are the matrices of queries/keys/values packed together:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \quad \text{with} \quad d_k = d_v = \frac{d_{model}}{h} \tag{2.7}$$

In addition to that, the attention used in the Transformer is *multi-headed*, which provides the attention with different representation subspaces at different positions:

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = concat(head_1, ..., head_h)\mathbf{W}^O, \tag{2.8}$$

with

$$head_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{2.9}$$

where h is the number of heads ($h = 8$ in the original Transformer), $\mathbf{W}_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_k}$ and $\mathbf{W}_i^V \in \mathbb{R}^{d_{model} \times d_v}$ are the projection matrices for queries, keys and values, which are used to project the input embeddings into a different representation subspace; the $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{model}}$ is a weight matrix that is used to condense concatenated attention heads into a single matrix. All three matrices are learnt jointly during the training.

Position-wise feed-forward neural network Each layer in both the encoder and decoder also contains a feed-forward neural network which is applied to each position separately. It consists of two linear transformations with a ReLU activation function in between. It transforms the output of the self-attention layer and produces the final output of the encoder layer.

$$FFNN(x) = ReLU(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{2.10}$$

Positional Encoding Positional encodings (PE) are used to provide the model with information about the relative or absolute positions of the words in the sentence. PE have the same dimensionality as the embeddings so that both can be summed. They are added to the input embeddings before they are passed to the encoder/decoder stack of the Transformer. The positional encodings are obtained from an encoding function that should be periodic, thus, the position of the token can be correctly represented, despite the length of the sentence. In the original Transformer Vaswani et al. utilized two sinusoidal functions of different frequencies:

$$\begin{aligned} PE_{(pos,2i)} &= \cos(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i)} &= \sin(pos/10000^{2i/d_{model}}) \end{aligned} \tag{2.11}$$

where pos is the position of given token and i is the embeddings dimension.

The positional encodings along with the attention-centric nature of the Transformer itself have changed the way in which the input is being processed. Now instead of presenting input sequentially, we are able to pass all tokens at once and get the information about their position from the embeddings.

2.5 Evaluation metrics

The purpose of automatic evaluation metrics for MT is to provide a quantitative measure of how accurate are the translations provided by an MT system without any assessment from a human translator. These metrics usually involve comparing the output generated by the MT system (*hypothesis* or *candidate* translation) to a *reference* translation. Two metrics that will be used for automatic evaluation in this work are *BLEU* and *chrF*.

2.5.1 BLEU score

The BLEU score [21] is a commonly used metric for evaluating the quality of the machine-generated text, particularly in the context of machine translation. The BLEU score compares a generated candidate translation to one or more reference translations.

The cornerstone of the BLEU metric is the *precision* measure, which is the proportion of *n-grams* (contiguous sequences of n words), that appear in both the candidate translation and the reference translation to n-grams that appear in candidate translation. The more n-grams that are shared between the candidate and reference translations, the higher the BLEU score.

$$p_n = \frac{\sum_{n\text{-}gram \in Cdt \cap Ref} count(n\text{-}gram)}{\sum_{n\text{-}gram \in Cdt} count(n\text{-}gram)} \tag{2.12}$$

It is computed as the geometric mean of the n-gram precision scores, where the n-gram precision for a given n is the *count* of n-grams in candidate translation that also appear in reference translation, divided by the total number of n-grams in candidate translation. Finally, the BLEU score is *brevity-penalized* (BP) if the candidate translation c is shorter than the reference translation r that has the closest length to c.

$$\textsc{Bleu} = \text{BP} \cdot (\prod_{n=1}^{N} p_n)^{\frac{1}{N}} \quad \text{with} \quad \text{BP} = \begin{cases} \exp(1 - r/c), & \text{if } c < r \\ 1, & \text{otherwise} \end{cases} \tag{2.13}$$

2.5.2 ChrF score

The chrF score proposed by M. Popović [22] is another commonly used MT evaluation metric. This metric was created to offset the problems identified during the period of using the BLEU metric. One of which is that the BLEU score measures the translation quality from a word-level perspective which makes it less accurate when it comes to the translation quality evaluation of morphologically rich languages.

The chrF metric is an F-score-based metric. To address the described issue the chrF compares hypothesis and reference on a character level instead.

$$\textsc{chrF}\beta = (1 + \beta^2)\frac{\textsc{chrP} \cdot \textsc{chrR}}{\beta^2 \cdot \textsc{chrP} + \textsc{chrR}} \tag{2.14}$$

where

CHRP is a character n-gram precision, the percentage of character n-grams in the hypothesis translation that are also present in the reference.

CHRR is a character n-gram recall, the percentage of character n-grams in the reference which are also present in the hypothesis translation.

β is a parameter that assigns β times more importance to recall than to precission.

In recent years, the chrF metric, along with its subsequent iterations – as chrF+(+) – has demonstrated a strong correlation with human translation quality estimation, particularly in the evaluation of translation quality for morphologically rich languages. To better illustrate this, let's consider the following example:

Source	**en**	Tired student was preparing for the exam.
Reference	**cs**	Unavený student se připravoval na zkoušku.
Hypothesis 1	**cs**	Unavená studentka se připravovala na zkoušku.
Hypothesis 2	**cs**	Unavený student se připravoval na zkoušku.

With no additional context provided both hypotheses listed above can be considered correct translations of the source sentence, the difference between them is that the first one contains the feminine forms of words and the second one - the masculine ones. Despite the fact that both translations are correct, the first one will reach a comparatively low BLEU score of 26.27 whereas the second one will reach a BLEU score of 100. ChrF metric offsets this problem by comparing the reference and hypothesis from a character-level perspective, the chrF score with $\beta = 3$ will be 0.74 and 1.0 for the first and the second hypotheses respectively.

Chapter 3

Approaches for low-resource neural MT

In this chapter, I will cover the contemporary techniques employed in NMT that are used when it comes to low-resource Machine translation.

3.1 Back-translation

Back-translation (BT) is the unsupervised training technique first proposed for SMT and then rediscovered for NMT by Sennrich et al. [3]. The idea behind this technique is that we can leverage the model's decoding ability and monolingual corpus during its training in a way of iteratively, for each observed training entry, first providing the perhaps incorrect translation of the source sentence by decoding and then learning to reconstruct the original sentence from this translated sentence. The inconsistency between these two sentences provides the error signal to train the model in the target-to-source direction.

3.2 Cross-lingual Language Model Pretraining

Cross-lingual language modelling (XLM) is a task of training language models that can predict tokens in multiple languages: language models trained with Masked language modelling or Translation language modelling (MLM and TLM) are able to predict the masked token at any position given the surrounding tokens (context), CLM – Causal language model – on the other hand, is able to predict a sequence of tokens after the preceding words. XLM is particularly useful in situations when the parallel data for a specific language pair is limited, or when the model's application involves the use of many languages.

Lample et al. [23] surveyed the capabilities of cross-lingual modelling to improve models' performance for various tasks – Cross-lingual classification, Cross-lingual language inference and Machine translation. In particular, to leverage XLM for MT, the Transformer's encoder is pre-trained using a causal, translation, or masked language modelling objective prior to being fine-tuned with the Machine translation (for parallel data) or Back-translation (for un-parallel data) objectives.

3.2.1 Language modelling

Language modelling is the NLP task that involves the modelling sequences of tokens in a given language. More specifically, the task involves training a model that predicts the probability distribution of the next word given the context of the preceding words. For models of this kind, the input will be the sequence of words and the output will be the probability distribution over the model's vocabulary for the next word. For example, for the English sentence

He was sitting in front of the _____

as the input, the output may be the distribution {..., "computer": 0.3, ..., "window": 0.4, ..., "charger": 0.0, ...} which signifies that the most probable next word is "window".

3.2.2 Masked language modelling

Masked language modelling (MLM) that was introduced in the BERT paper by Devlin et al. [15] is a type of language modelling that, unlike the previously described approach, has the ability to predict the word at any position. As it can be seen in Figure 3.1, training of the masked language model consists of randomly masking a few tokens in the input sentence (15% of tokens in the original paper) and then predicting the word that was in the original unmasked sentence based on surrounding tokens. The masking itself is simply a replacement of a chosen word with $[MASK]$ token 80% of the time, a random token from the vocabulary 10% of the time and an unchanged token 10% of the time.

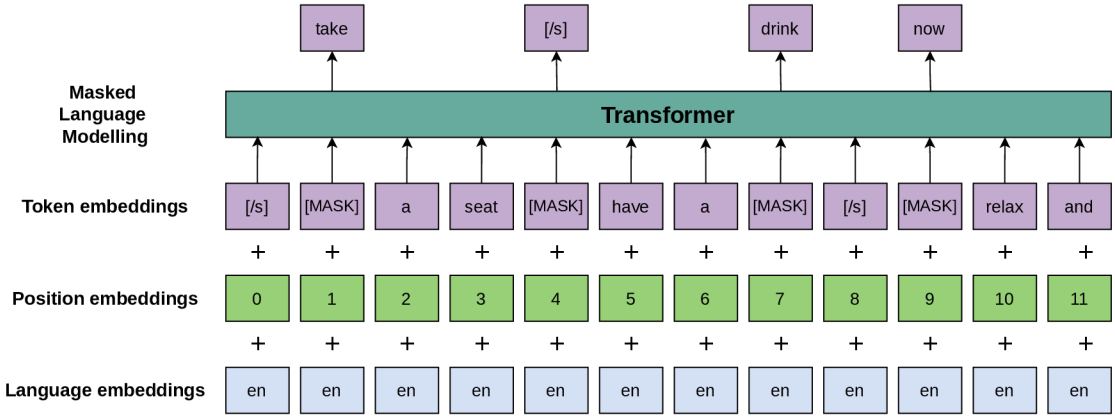


Figure 3.1: XLM variant of MLM training, $[\backslash s]$ tokens denotes the boundaries of text streams. Replicates the scheme from [23].

Lample et al. modified the original training algorithm by using the input streams of an arbitrary number of sentences instead of pairs of sentences. Also, in order to counter the imbalance between rare and common tokens they discarded the frequent words in the training set with the probability given by the formula

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \tag{3.1}$$

where $f(w_i)$ is the frequency of the word w_i and t is a threshold parameter.

3.2.3 Translation language modelling

In order to further improve the cross-lingual pretraining Lample et al. proposed a novel language modelling objective called *Translation language modelling*. TLM, being an extension of MLM, tries to leverage parallel data.

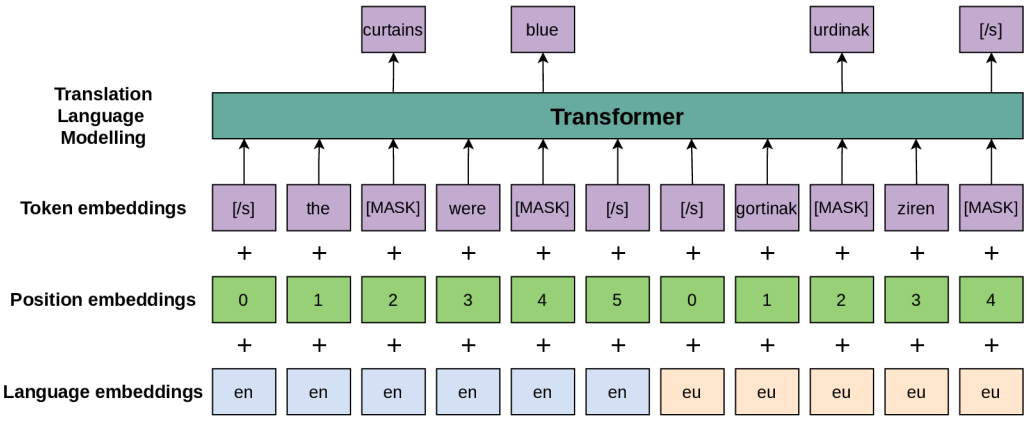


Figure 3.2: Training the English-Basque LM with translation language modelling objective.

As this can be seen in Figure 3.2, the training with such an objective consists of a concatenation of parallel sentences, masking random tokens (with no regard to tokens' language) in the new concatenated text and then predicting the original tokens that were masked based on tokens surrounding masked tokens in both source and target languages. It is also worth mentioning that the positions of tokens in the target sentences are reset. Training model this way encourages it to learn the alignments between the representations of parallel sentences.

The TLM pretraining proved to be useful for the Cross-lingual classification task, the aforementioned research shows that leveraging data through the TLM+MLM objective provides a boost in performance of 3.6% accuracy.

3.3 Leveraging bilingual dictionaries for cross-lingual pretraining

"The unsupervised training techniques, based on a bilingual dictionary, as presented by Duan et al. in [1], rely on the use of unparallel corpora in at least two languages, along with a bilingual dictionary that provides word mappings. In this section, I will describe these training methods.

3.3.1 Bilingual dictionaries

Bilingual dictionaries (or lexicons) are collections of pairs of mutual translations of expressions (single words or phrases) in two languages. The main benefit that bilingual dictionaries bring is that they may provide multiple ground-truth translations of a single expression that could not usually appear in datasets. However, bilingual dictionaries often provide translations without any additional information regarding the translation pair: word class (part of speech) and the case- or tense-related information may be omitted. Another downside

is that we can't extract the context in which a given word is usually used neither on the source nor the target side.

3.3.2 Anchored training

Target view Anchored training To bridge the gap between semantic spaces of languages Duan et al, considering the fact that the cross-lingual language model is being trained on monolingual corpora and that word embeddings are highly dependent on a word's surrounding context, proposed a method called Anchored training (AT) which leverages the bilingual dictionary by injecting *anchors* on the monolingual corpora.

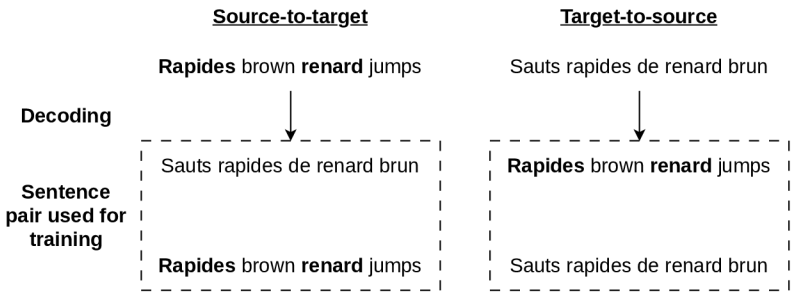


Figure 3.3: Example of Back-translation used for AT made in both directions for the same sentence pair, anchor words are written in bold.

As it is evident from Figure 3.3, the essence of the AT method is that we, having a monolingual corpus and bilingual lexicon, replace the words covered by this lexicon with *anchors* – translations of these words provided by the lexicon. Through these anchors, words with similar meanings in different languages will share the same word embeddings. The training process consists of the back-translation steps in both – source-to-target (s2t) and target-to-source (t2s) – directions:

1. The anchored source sentence is translated via s2t NMT decoding into the target sentence and then two of them are used as a pair for t2s translation training.
2. The target sentence is translated into the anchored source sentence via t2s decoding and then both sentences are used to train the s2t translation model.

During the inference, the source sentence is also in advanced translated into the anchored sentence by looking up the dictionary and only then the s2t decoding is used.

Bi-view Anchored training The modification to AT, the Bi-view AT, extends the way the model looks at the sentence pair by adding the source language view to the target language view that was described previously. This extension provides two additions: first, we add steps similar to those previously described for AT, with the exception that now substitutions are applied to target sentences; second, usage of a dictionary in a bidirectional manner allows us to generate additional anchored sentences from true denoised sentences produced by decoding in either of 2 directions, which will be also utilized during the training.

3.3.3 Anchored Cross-lingual pretraining

Another method presented in this article is *Anchored Cross-lingual pretraining* (ACP). It assumes the absence of parallel corpora. ACP involves the same noising step that was

previously described for AT – we first generate the pseudo-parallel sentence based on the source sentence by substitutions of words covered by a bilingual dictionary.

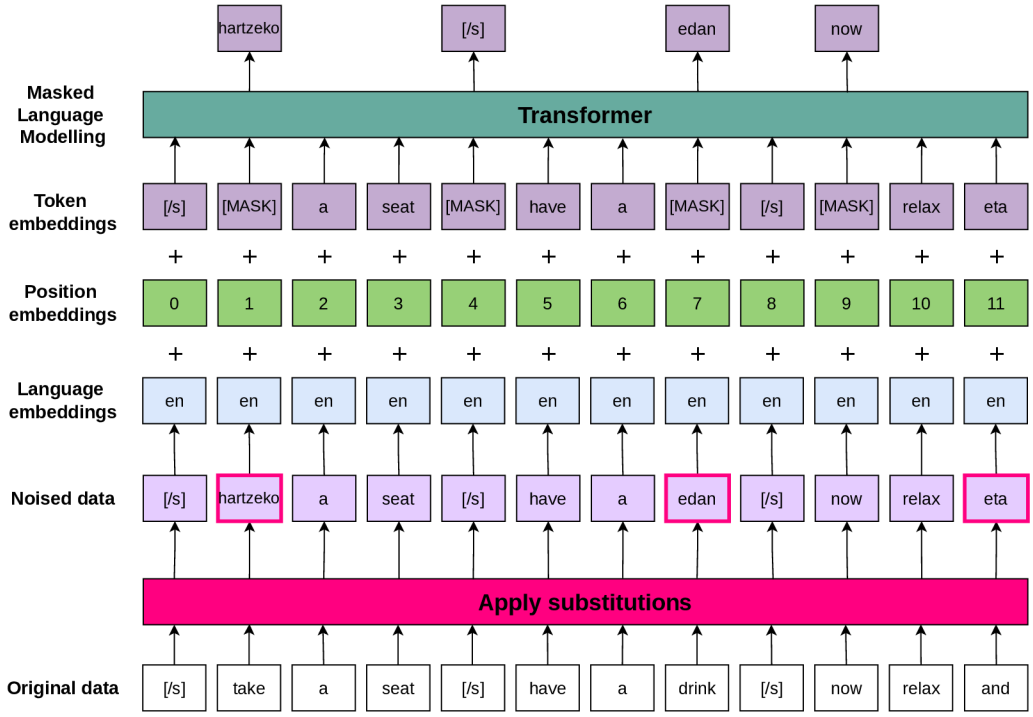


Figure 3.4: Anchored Cross-lingual Pretraining – leveraging noisy data through MLM objective.

As can be seen in Figure 3.4, the source words are replaced with their target language counterparts, and the resulting noised sentence, treated as if it was in the source language, is leveraged through the Masked language modelling objective.

3.4 XLM with pseudo-parallel data

Another method that is covered in this work includes the use of pseudo-parallel data. As can be seen in the scheme from Figure 3.5, we, having only the monolingual data in one language are generating the noisy version of it. Then we are, treating the true source corpus and the noisy version of it as the source data and the target data respectively, leverage them through the language modelling objective (either TLM or MLM).

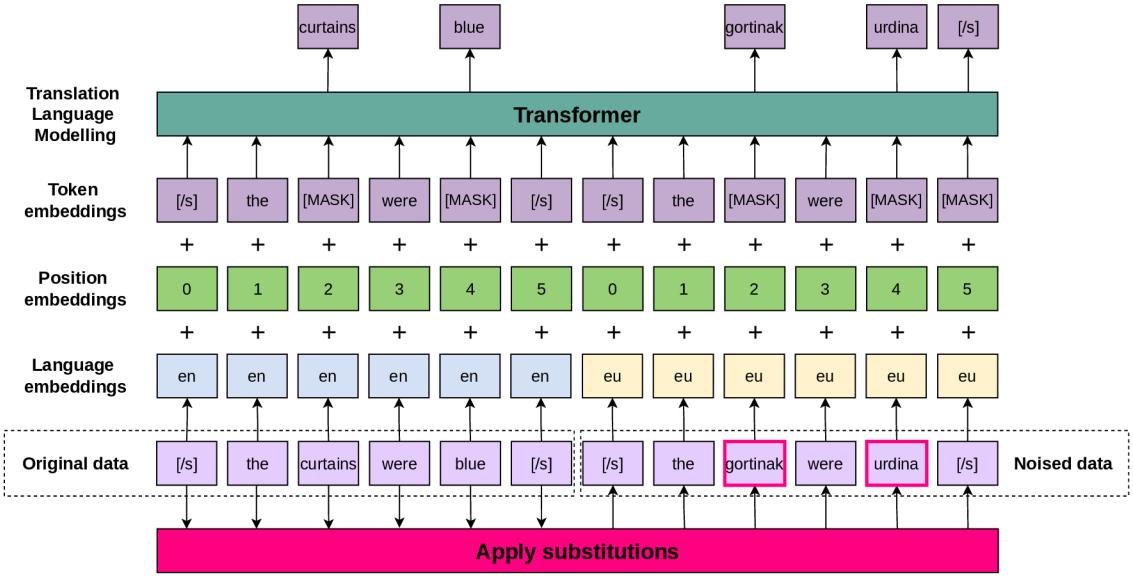


Figure 3.5: XLM on pseudo-parallel data with TLM objective.

Chapter 4

Implementation

The XLM framework[1] provides training options for MLM and TLM pretraining as well as training options for Machine translation and Back-translation training out-of-the-box.

However, the ACP and AT techniques are not directly embedded in the XLM framework. In order to leverage the data the way these methods expect to, we have to modify the data used during the training in the following manner for each of the methods and then leverage them through corresponding training objectives:

ACP assumes the unsupervised training on non-parallel corpora, where the source side corpus is noised with anchoring points and the target side corpus remained clean. Data are leveraged through the MLM objective.

AT assumes the unsupervised training through the BT objective in both directions (source-to-target and target-to-source) on monolingual corpora with only the source side corpus noised by placing the anchors provided by the bilingual lexicon.

XLM on pseudo-parallel data assumes that the target language data is generated by the BD-based substitutions applied to true source data. The data will be leveraged through MLM or TLM objectives as it is depicted in Figure 3.5.

The script that applies the substitutions is named `sub_by_dict.py`, the substitution algorithm itself is described in Appendix A.1. All the training arguments used for all the experiments described further are listed in Appendix C.

[1] https://github.com/facebookresearch/XLM

Chapter 5

Experiments

In this chapter, I will cover the subjects closely related to the experiments conducted for this work: used data, data preprocessing pipeline and rationalization of the choice of languages.

Later parts of this chapter analyse the effectiveness of the application of described bilingual dictionary based techniques under different conditions. This work investigates the settings that are not covered by the research of [1] – first, I surveyed the situations which involve the use of a small amount of parallel data and various amounts of non-parallel data; second, a case somewhat similar to the one discussed in the above-mentioned study, with the exception that there will only be a limited amount of monolingual data for both languages.

5.1 Data overview

In this section, I would like to list and briefly describe all data sources that were used to conduct the experiments. In subsections 5.1.1 and 5.1.2 I will review in terms of data quality, domain and size the corpora that were used as training, validation and testing sets during the models' training and evaluation. In subsection 5.1.3 used bilingual dictionaries are covered. Unless otherwise specified, all listed corpora were downloaded from OPUS [24].

5.1.1 Training datasets

CCMatrix [25] is the giant corpus containing parallel sentences in 90 languages. The content of it was extracted by data mining from web crawls obtained from the entire web. The implication of that is the fact that the content may contain many noisy entities: hyperlinks, random punctuation, markup elements, and misalignments.

Europarl [26] is a parallel corpus containing 21 European languages. It consists of transcripts of European Parlament sessions translated into these languages by professional translators.

NewsCommentary is a parallel corpus that was created by WMT. It consists of news commentaries and is available in 15 languages.

Wikimedia is a large collection of parallel sentences in 322 languages. It includes the dumps of articles from various Wikimedia projects (Wikipedia, Wiktionary and others).

	Size of datasets (in sentences)				Domain
	en-fr	en-cs	en-vi	en-eu	
CCMatrix	-	-	-	7.8M	Various
Europarl	2.1M	647K	-	-	Politics
NewsCommentary	-	212K	-	-	News
Wikipedia	818K	-	58K	-	Wiki articles
Wikimedia	1.0M	-	-	-	Wiki articles
OpenSubtitles	-	-	350K	-	Movie subtitles
ElhWebCorp	-	-	-	12M	Various
EhuHac	-	-	-	585K	Various
TED2020	-	-	326K	-	Various
NeuLab_TedTalks	-	-	183K	-	Various
EUBookshop	10.8M	-	-	-	Legal
QED	-	-	338K	-	Various
flores-200_dev	997	997	997	997	Various
flores-200_devtest	1012	1012	1012	1012	Various
Tatoeba	268K	31K	5694	2066	Various

Table 5.1: Overview of all datasets that are used in this work.

OpenSubtitles is a parallel corpus of translated movie subtitles from OpenSubtitles web site[1] it is available in 62 languages.

ElhWebCorp (Elhuyar Basque Web Corpus) is a product of Igor Leturia's Ph.D. thesis [27]. It was collected using both search engines and crawling. It is a large monolingual Basque general corpus containing around 186M raw tokens. As well as the CCMatrix corpus, ElhWebCorp includes a lot of noisy elements.

EhuHac is a dataset created by Basque Country University. The corpus is built on translations of 171 books, it provides parallel data with Basque on one side and French/English/Spanish on the other.

TEDTalks NeuLab-TedTalks[2] and TED2020 are both corpora that are based on transcripts of TED Talks presentations. TED2020 was made by Reimers and Gurevych [28], it is based on translated subtitles of 4000 TED talks in 100 languages. NeuLab-TedTalks is based on translated subtitles that were made by volunteers, the dataset is available in 59 languages.

EUBookshop is a parallel corpus that was created by crawling the EU Bookshop, the archive of various publications from EU institutions. The dataset is available in 48 languages.

QED (QCRI Educational Domain Corpus) is a parallel corpus that consists of translated subtitles of educational videos and lectures developed by Qatar Computing Research Institute [29]. The corpus supports 225 languages.

[1] http://www.opensubtitles.org/
[2] https://github.com/UKPLab/sentence-transformers/blob/master/docs/datasets/TED2020.md

5.1.2 Validation and test datasets

Flores-200 [30] is a high-quality parallel multilingual benchmark for low-resource and multilingual machine translation. Flores-200 consists of translations of 842 distinct web articles for 200 languages made by professional translators, the dataset is divided into two splits: *dev* and *devtest* that were used as validation and test sets respectively for each language pair in this research. Flores-200 benchmark can be downloaded from the flores200 GitHub repository[3].

Tatoeba [31] is a large corpus that is based on the Association Tatoeba database of sentences and their translations that are being contributed by the community. The main product of the project is the toolkit that provides the examples of usage of given words. And thus the parallel corpus consists mostly of simple "example sentences" in different languages. Tatoeba is available in 380 languages. Tatoeba corpora were used for model testing for all declared language pairs.

5.1.3 Bilingual dictionaries

MUSE dictionaries are the high-quality bilingual dictionaries for 110 languages that were presented alongside a state-of-the-art approach for unsupervised MT in the work of Conneau et al. [32]. Authors took into account the problem of words' polysemy and instead of using some online translation tools to generate expressions' translations they used an internal translation tool designed specifically for the task of lexicon induction. In this work, Vietnamese-, French-, and Czech-English MUSE dictionaries are utilized. Bilingual lexicons are publicly available as part of the MUSE library[4].

ELRC English-Basque dictionary is an automatically created dictionary that was inferred during the training of an unsupervised MT system as a part of MT4ALL project[5].

Apertium [33] is an open-source platform for rule-based non-neural machine translation. One of the modules of this system, namely, lexical transfer, relies heavily on a bilingual dictionary for looking up lexical forms in them. Unlike other bilingual lexicons, the ones used in Apertium systems provide additional information like the word's part of speech, case, and tense to further improve the language translation. However, in this research dictionaries will be only used for word mapping. English-Basque bilingual lexicon from the research of O'Regan and Forcada [34] is utilized in this work.

It is worth noting that the resulting English-Basque lexicon has lower quality compared to the ones provided by the MUSE library, it includes many entries with translation of phrases which may substantially reduce the dictionary coverage of the dataset. Also, the vocabulary was pre-processed by cleaning from noisy elements (redundant punctuation, unrelated symbols).

[3] https://github.com/facebookresearch/flores/blob/main/flores200/README.md#download
[4] https://github.com/facebookresearch/MUSE
[5] https://ixa2.si.ehu.eus/mt4all/project.html

5.2 Languages

In order to better explore the capabilities of bilingual dictionary based techniques it was decided to train the MT models for several language pairs. All of them consider English as the source language. The choice of target languages is based on their morphological richness and linguistic distance from English.

Here is the list of these target languages and some of their linguistic features that are relevant to this research:

French is a Romance language that is very closely related to English. French and English share about 30% of words that are either derived from Latin or borrowed by one language from another. Just like English, French has the SVO word order. French was chosen as a language with a small linguistic distance from English and as a language of a comparably simple morphology.

Vietnamese is an isolating language, which means that words are made up of morphemes that cannot be further divided into meaningful smaller units. Similar to English, it has an SVO word order. Vietnamese can be characterized as a language with low morphological complexity: it lacks any direct modifications of words. Instead, to indicate any grammatical inflection, additional words are added before or after the word we want to inflect. Because of the combination of all these properties, it is expected that the performance boost provided by BDBNMT for English-Vietnamese MT will be the highest among all considered language pairs.

Original English Data	There are schools but there is no paper.
Vietnamese translation*	Có trường nhưng không có giây.
Generated sentence	There đang trường nhưng kia quan không trát.
Translation of gen. sentence*	There's school, but there's no warrant.

As one can see in this example, the translation generated by substitution is not very accurate: it is verbose and English-centric due to the word mapping in the English-to-Vietnamese direction. However, it allows the model to learn better representations of Vietnamese words during the pretraining phase: in the case of this sentence, having the vocabulary entry {"paper": ["trát", "giây"]} with two Vietnamese translations, we, using the substitution algorithm, get *trát* ("warrant") as a translation, which is not accurate in this situation but these two words have a somewhat close meaning and thus it is desirable that they are located close to each other in the embedding space.

Basque is an under-resourced language which is considered a language isolate (there are no other languages related to Basque). Basque is known for its complex morphology and agglutinative nature. It has a relatively flexible word order, which means that words in a sentence can be rearranged for emphasis or stylistic purposes without really changing the meaning of the sentences. However, the existing rules imply that the basic word order in Basque is SOV (subject-object-verb). To better understand the

*Translation generated by Google Translate

morphological richness and agglutinative nature of the Basque language let's consider the example*: the verb *dakartzat* ("I bring them") is formed as follows: *da* indicates present tense, *kar* is a root word of infinitive *ekkari* ("to bring"), *tza* indicates plural and *t* indicates subject ("I"). Such high morphological complexity causes doubts about the applicability of bilingual dictionary based word substitutions: lexicons simply can't provide mappings to all the forms that are inflected from the word. However, considering the fact that Basque is an under-resourced language, BD-based can still be effective for improving the model's performance by the provision of ground-truth expressions' translations.

Czech is a fusional language which implies that the meaning of a word is often changed by adding suffixes, prefixes, or inflections to the base word. Czech nouns, adjectives, pronouns, and verbs are inflected to reflect their grammatical role in a sentence, as well as their gender, number, and case. Even though Czech, like many other Slavic languages, has a relatively free word order, in general, it follows the SVO word order.

5.3 Data preprocessing

The preprocessing of corpora in general follows the same steps that are shown as the example in the XLM GitHub repository:

1. Tokenize the data using the rule-based tokenizer which will split words into subwords and separate the interruption.
2. Learn BPE codes from both source and target tokenized data.
3. Apply the learned BPE codes on tokenized data.
4. Extract the vocabulary from BPE tokenized data.
5. Divide the data into parallel and non-parallel subsets if we expect to train the model with TLM objective and there are non-parallel segments.
6. Binarize the data.

5.4 Tools

Preprocessing To perform data preprocessing described in Section 5.3 the following tools were used:

Moses tokenizer[6] is used for subword tokenization of raw data.

fastBPE[7] is used for all BPE-related operations (BPE codes learning, vocabulary extraction, BPE codes application)

Training XLM[8] framework was used for model training with both Language modelling and Machine translation objectives.

*Taken from Wikipedia

[6] https://github.com/moses-smt/mosesdecoder

[7] https://github.com/glample/fastBPE

[8] https://github.com/facebookresearch/XLM

Evaluation In order to perform models' performance evaluation during both validation and testing the following tools were used:

Since the XLM framework provides no option for the training with the chrF metric, the evaluation module was modified by adding a chrF score implementation provided by nltk [35] and a corresponding option for a training script.

SacreBLEU implementation of chrF and BLEU scores is used for the evaluation of converged MT models, sacreBLEU was introduced by M. Post [36].

5.5 Basic sets: parallel data for training MT systems

In order to provide some parallel data for MT training, subsets of 130k sentences dubbed *basic sets* were extracted from the following corpora: Europarl, NeuLab-TedTalks, Europarl, CCMatrix for English-French, -Vietnamese, -Basque and -Czech language pairs respectively. Basic sets are the only parallel datasets used in this research. They are used in all experiments that assume the use of parallel data.

5.6 Notation

Throughout the following experiments' descriptions and discussions, some specific abbreviations and terms will be used. The list is as follows:

Training objectives (MLM, TLM, MT, ACP, AT) will be denoted as an abbreviation with a subscript indicating the number of training sentences from both source and target sides, the prime sign will mean that data used for training are pseudo-parallel, stages of training are separated by a comma. E.g. TLM100+MLM130′,MT130 means that the model was first pre-trained: with TLM objective on first 100k sentences and MLM objective on the next 130k pseudo-parallel sentences; next, the model was MT fine-tuned leveraging only 130k sentences.

MT Machine Translation.

BD Bilingual Dictionary.

T-/M-LM Translation/Masked Language Modelling.

ACP Anchored Cross-lingual Pretraining.

AT Anchored training.

5.7 Training models from scratch with a small amount of additional data

This set of experiments emulates the situation where one doesn't possess the large-scale corpora to train an NMT model. The training was conducted for all 4 language pairs that were listed previously. These experiments should explore the efficiency of models' pretraining applicability of the BD-based approach to languages of different morphological complexity. It is expected that the performance boost may be less evident for more morphologically rich languages.

5.7.1 Determining the optimal hyperparameters

Since it is expected that the performance of MT models will be subpar because of data scarcity, it was decided to find the embedding dimensionality that will be optimal for these models. Related experiments are described in Appendix B.

5.7.2 Baseline

The baseline case considers the absence of additional monolingual data, so baseline models are leveraging only the basic sets for training. Experiments conducted will also serve as a showcase of how efficient both TLM and MLM pretraining with following MT fine-tuning are compared to the pure MT approach and to one another.

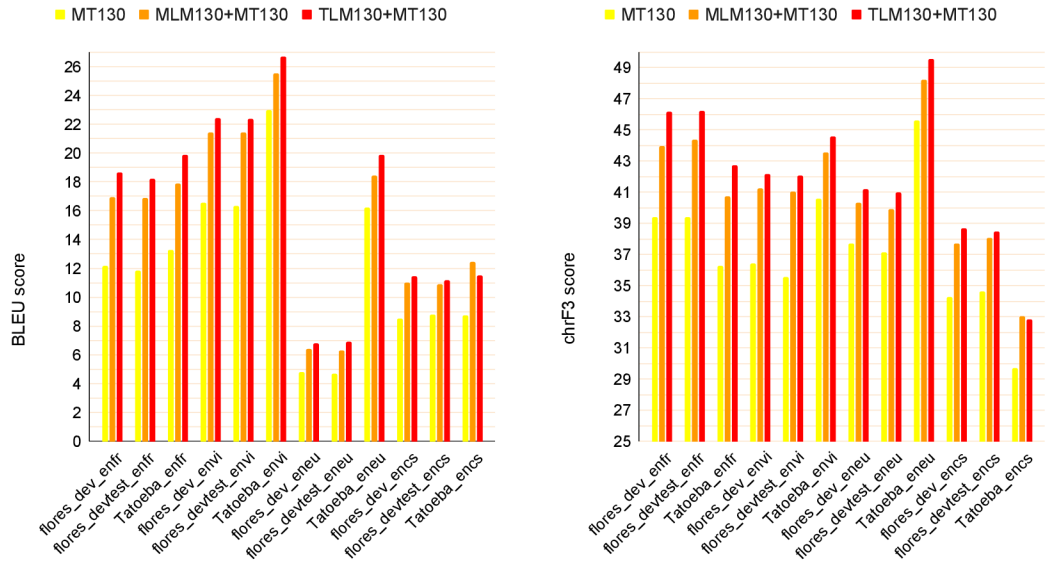


Figure 5.1: Comparison of performance of models trained without pretraining and models trained with different pretraining objectives.

First, in order to better understand the significance of LM pretraining it was decided to train models with the MT objective without any pretraining to provide a point of reference. As it is apparent from both graphs from Figure 5.1, MLM pretraining provides a notable performance boost in the BLEU score (30%) and in the chrF3 score (10%) on average among all the test sets.

Second, to explore the expediency of leveraging parallel data through the TLM objective was trained the MT model that was pre-trained with TLM. Graphs from Figure 5.1 demonstrate that MT models that were first TLM pre-trained outperform models that had MLM pretraining for each language pair. To enhance the performance of future models, it was decided to always consider the basic set suitable for the TLM objective.

Test sets' complexity

From the graphs in Figure 5.1 it is evident that the Tatoeba set is different from flores200 dev/devtest sets in terms of complexity – as mentioned before, Tatoeba set consists of simple short example sentence whereas flores200 is based on web articles and consists of

sentences that are more complex, so it is worth considering that results for flores200 sets are in general more representative. For all subsequent experiments, graphs will only depict the average performance for a particular model over all datasets. The summary of the performance of every model trained during this research for each dataset can be found in Appendix D.

5.7.3 XLM with the use of pseudo-parallel data

In order to provide a set of English monolingual data to be used for BD-based substitutions to generate the pseudo-parallel data, 100k sentences were extracted from the corpora listed in Table 5.2.

Language pair	Corpora	BD entries	BD coverage
English-French	English-French Europarl and Wikipedia	113286	54.6%
English-Vietnamese		76364	49.8%
English-Basque		29282	35.5%
English-Czech	English-Czech Europral and NewsCommentary	64211	54.7%

Table 5.2: Information about corpora that were utilized as additional data as well as the percentage of words covered by bilingual dictionary.

After the application of BD-based substitutions on monolingual corpora we obtain 100k pseudo-parallel sentences. After the merge with the true parallel basic set, we obtain a parallel corpus of 230k sentences. At the pretraining stage, the first 130k will be utilized for the TLM objective, the remaining 100k will be utilized for the MLM objective, and then the model will be fine-tuned with the MT objective on the true parallel part.

5.7.4 Topline

To train models that suppose to provide better (topline) results compared to models that were trained on pseudo-parallel data, this time non-synthetic data were utilized as the target language data and were labelled as non-parallel.

The additional data for the target side of training sets for each language were taken from the following corpora (50k sentences from each listed corpus): English-French – Europarl, Wikipedia; English-Vietnameses – NeuLab-TedTalks, OpenSubtitles, English-Basque – ElhWebCorp, EhuHac, English-Czech – Europral, NewsCommentary.

5.7.5 Results analysis

From the performance graphs in Figure 5.2 it can be clearly seen that the addition of pseudo-parallel data leads to the substantial improvement of models' performance for language pairs with a high BD data coverage of around 50% (English-Frech, -Vietnamese, -Czech). Models that were pre-trained using XLM with the use of pseudo-parallel data demonstrate either a similar or superior performance compared to the models marked as Topline which were leveraging the true monolingual additional data for pretraining.

Another observation is the fact that for the English-Basque model, the lower BD data coverage and lower quality of BD lead to limited improvement in the resulting MT model's performance.

In conclusion, it can be said that the use of pseudo-parallel data generated by a large BD for language modelling can be extremely beneficial for the resulting MT models' performance under the described setting – up to the point that models pre-trained on this kind of data outperform the ones that were trained on true non-parallel data.

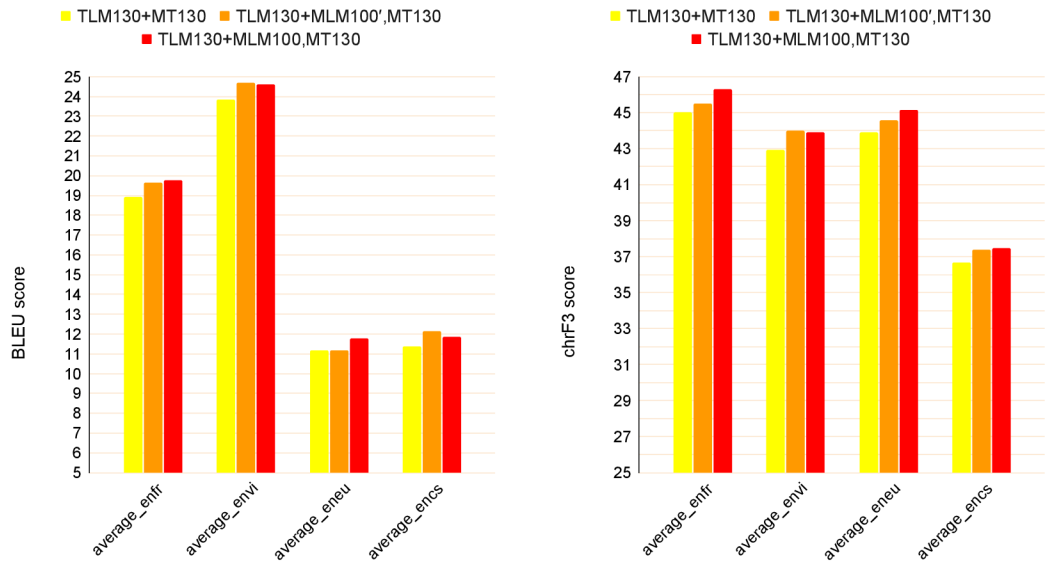


Figure 5.2: Performance comparison of Baseline/XLM with pseudo-parallel data/Topline models.

5.8 Training models from scratch on a small amount of monolingual data

Unsupervised methods of training such as the BT require an enormously big amount of unparallel data, usually several millions of sentences on source and target sides. BD-based methods – ACP and AT – proposed by Duan et al. [1] demonstrated their efficiency in the setting of an abundance of monolingual data.

This series of experiments suppose to examine the effectiveness of the proposed training method in the setting of a scarce amount of monolingual data.

5.8.1 Baseline

Since it is expected that the performance of the resulting model trained with ACP+AT training steps will be extremely low. The translation made by the application of a bilingual dictionary will be considered a Baseline performance (basically, the noised versions of original corpora are used as the hypothesis translations).

5.8.2 Anchored training

For the ACP+AT training were used small-scale non-parallel corpora of 230k sentences. The source language corpus consists of pseudo sentences generated from the true source language data.

5.8.3 Results analysis

As one can see from the resulting performance depicted in the graph from Figure 5.3, ACP+AT training on a small-scale monolingual corpora shows translation quality that is inferior to the one provided by the word mappings from bilingual dictionaries.

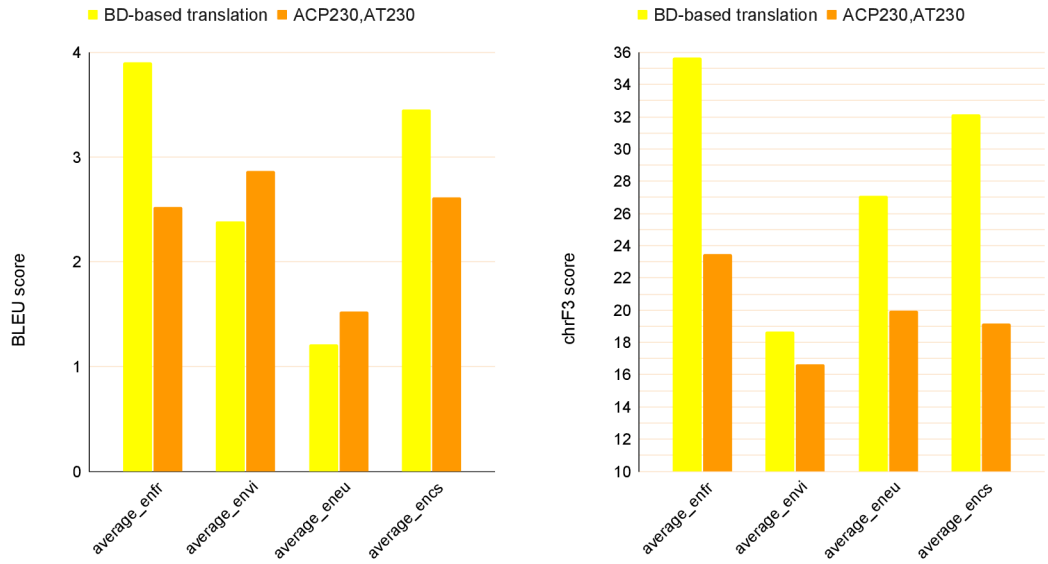


Figure 5.3: Transaltion quality comparison of translations provided by bilingual dictionary and by the ACP+AT trained model.

The deteriorating performance of the trained model can be explained by the fact that unsupervised methods of training require large amounts of data, ACP and AT methods proved to be efficient exclusively under the setting of the abundance of non-parallel data.

5.9 Fine-tuning the ACP+AT pre-trained model

Since all other experiments for model pretraining only consider the tuning of the Transformer's encoder, it was decided to test the performance of the MT fine-tuned model the pretraining of which includes the training of a decoder. Models described in Section 5.8 trained solely on monolingual corpora suit this setting since they had a stage of AT training, which includes the optimization of parameters of a decoder.

5.9.1 Baseline

Since this series of experiments consider the usage of parallel data for MT fine-tuning, we can utilize results provided by MT-tuned models from previous experiments for translation quality comparison. As the Baseline model will be considered the model that was MLM pre-trained on the same data as the model described in Section 5.8 and then fine-tuned on a basic set.

5.9.2 MT fine-tuning

Since the preceding ACP and AT training procedures were done on source data noised with BD-based substitutions, it is not obvious whether the models should be fine-tuned with the MT objective on data with the noised or the clean source side.

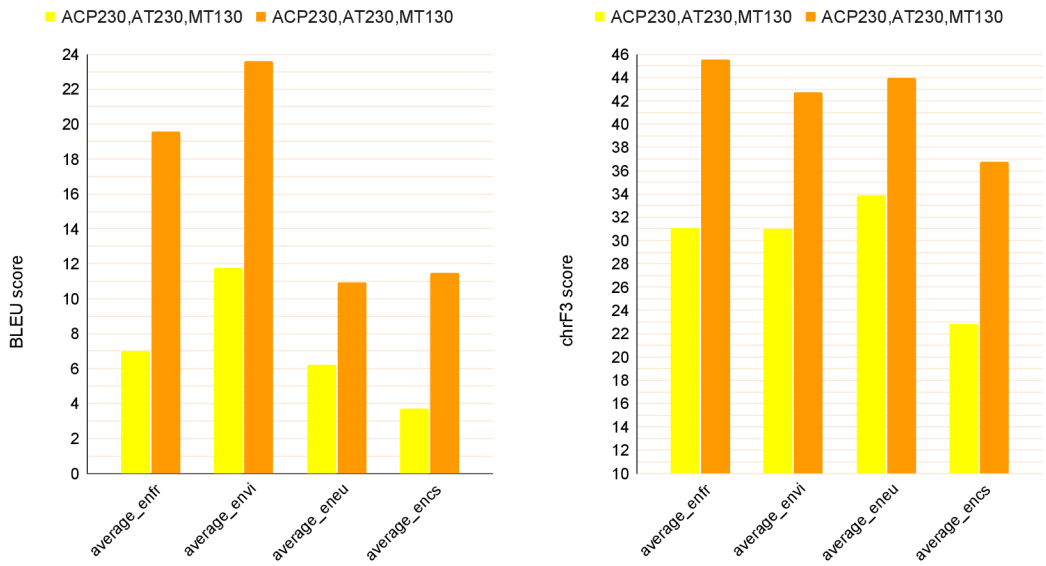


Figure 5.4: Comparison of performance of models which were fine-tuned with MT objective on noised source data (yellow) and clean source data (orange).

Performance graph from Figure 5.4 shows that the models that were MT fine-tuned on parallel data with clean source side outperform ones that considered noised corpus as a source language data. The possible explanation for such degrading performance may lie in the fact that the substitutions that were observed during the training didn't appear in the test data.

5.9.3 Topline

The performance of TLM pre-trained Topline models described in 5.7 will be considered a Topline performance.

5.9.4 Results analysis

From the testing results depicted in Figure 5.5 it can be seen that the ACP+AT pre-training yields results that are either similar or superior to the ones delivered by the systems which were first pre-trained with MLM objective.

It can be said that tuning the Transformer's decoder parameters during the pre-training via ACP+AT training may be beneficial in some cases – there is a significant improvement for the model for the English-French language pair, however, the improvement yielded by such pre-training is not persistent for different language pairs, so this method of pre-training cannot be considered superior to the MLM.

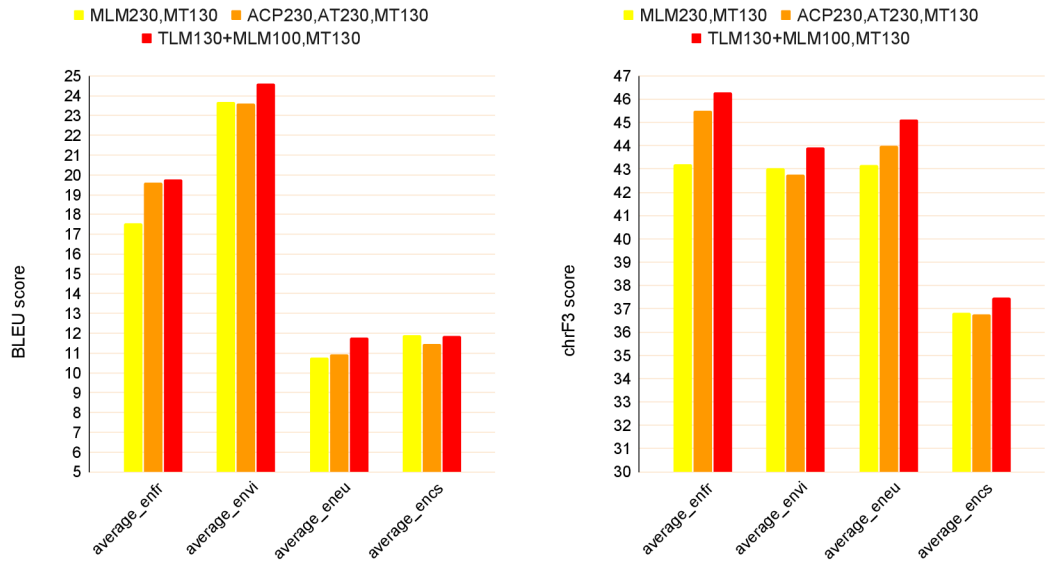


Figure 5.5: Performance comparison of Baseline/ACP+AT pre-trained/Topline models.

5.10 Training models from scratch with a large amount of additional data

This series of experiments suppose to investigate the applicability of the BD-based methods to the situation where we add a large amount of monolingual data for the model to leverage in addition to previously described parallel basic sets.

Models will be trained for two language pairs with comparatively low morphological complexity: English-French – suppose to show how efficient pretraining on pseudo-parallel data is for linguistically close languages, English-Vietnamese – suppose to show how efficient it is efficient for distant languages.

Since the amount of training data was increased, the size of embedding vectors utilized by models trained for this series was increased to 512 to enhance the models' capacity to represent the language's features.

5.10.1 Baseline

The Baseline case considers the presence of large-scale monolingual corpus only for the source side. This way the model will have a lot of data to generalize on which may lead to the emergence of bias towards the source language. However, due to the fact that the additional data are provided for the source side, it may not be as noticeable since the model will have a good ability to represent the source language sentences and thus the ability to translate them to sentences in the target language.

5.10.2 XLM with the use of pseudo-parallel data

Having the same initial conditions as the ones described for the Baseline, we now extend the training corpus by generating the noisy data that will serve as target language data.

	Number of words	Number of substitutions	Coverage
English-French	24713580	13408303	54.2 %
English-Vietnamese	18869018	9472170	50.2 %

Table 5.3: Dictionary coverage for each language pair.

5.10.3 Topline

The Topline case resembles a situation somewhat close to the one investigated by Duan et al. in [1]. This case assumes the presence of large-scale non-parallel corpora – 1M sentences for each language, so the language modelling training is done on true data.

5.10.4 Results analysis

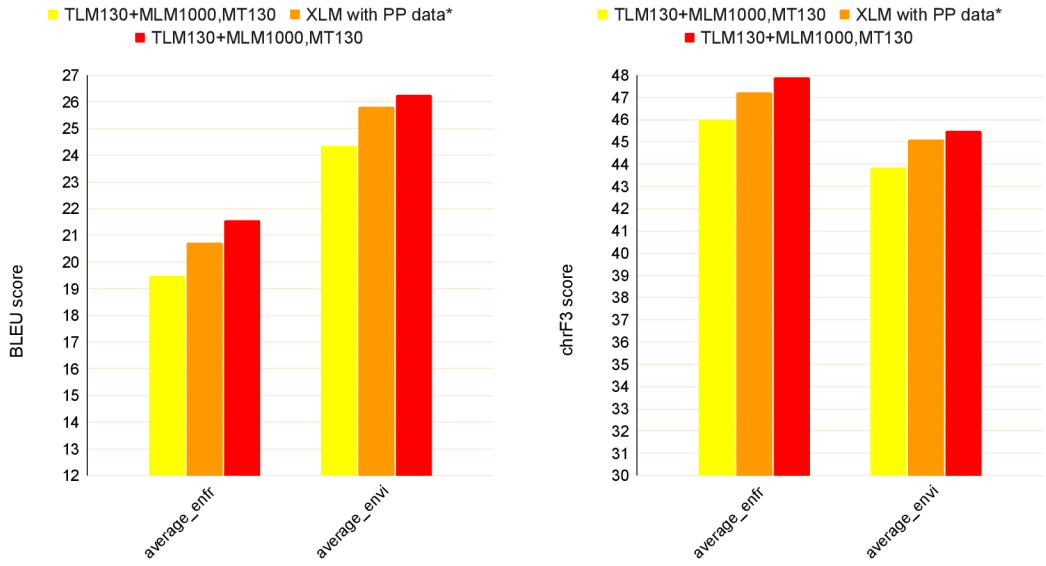


Figure 5.6: Performance comparison of Baseline/XLM with pseudo-parallel data/Topline models. Scores marked with an asterisk correspond to the performance of models from Section 5.11 that are best for a particular language pair.

From the performance evaluation depicted in graphs from Figure 5.6 it is evident that XLM models trained with the use of pseudo-parallel data yield a significant boost in performance compared to the models that were utilizing solely additional source data.

The initial assumption that the performance of the improvement provided by the addition of (pseudo-)parallel additional data models will depend on the linguistic distance proved to be wrong: the difference between the improvement yielded by Topline models is not significant between two language pairs – 10.7% and 7.76% BLEU score improvements for English-French and English-Vietnamese respectively. What is observable, however, is the fact that the boost provided by the inclusion of additional pseudo-parallel sentences is more significant for the English-Vietnamese language pair than for the English-French lan-

guage pair which may indicate that the target language's morphological complexity plays a significant role in whether this method will be useful for a particular language pair or not.

In conclusion, it can be stated that XLM with the use of pseudo-parallel data is a viable approach for enhancing the models' performance in this setting. However, unlike for the experiments with a small amount of additional monolingual data (Section 5.7), it cannot be stated that this method provides results that are similar to ones obtained from training on true monolingual data since there is a noticeable performance gap.

5.11 Applicability of TLM pretraining on pseudo-parallel data

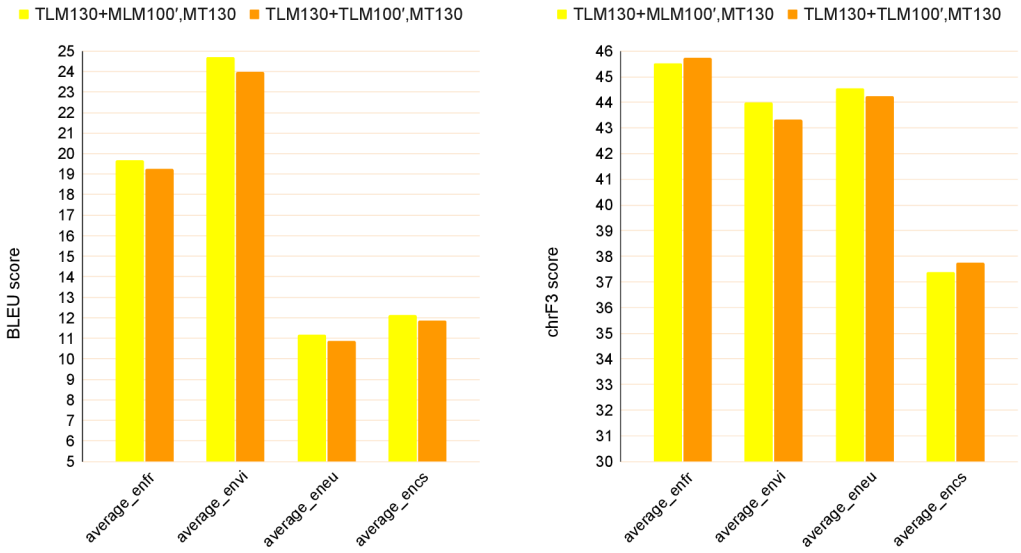
As previously mentioned, when it comes to the utilization of parallel corpus through some language modelling objective for further MT fine-tuning the TLM yields better results than the MLM. However, it is debatable whether we should consider the pseudo-parallel data generated by the BD-based substitutions parallel or not.

For the experiments described in Sections 5.7 and 5.10, there were two groups of models that were trained with the use of additional pseudo-parallel data. Models dubbed *TLMcat* were considering pseudo-parallel data as parallel and thus were leveraging them through the TLM training objective. Models dubbed *TLMsep*, on the contrary, treated pseudo-parallel data as available exclusively for the MLM training objective.

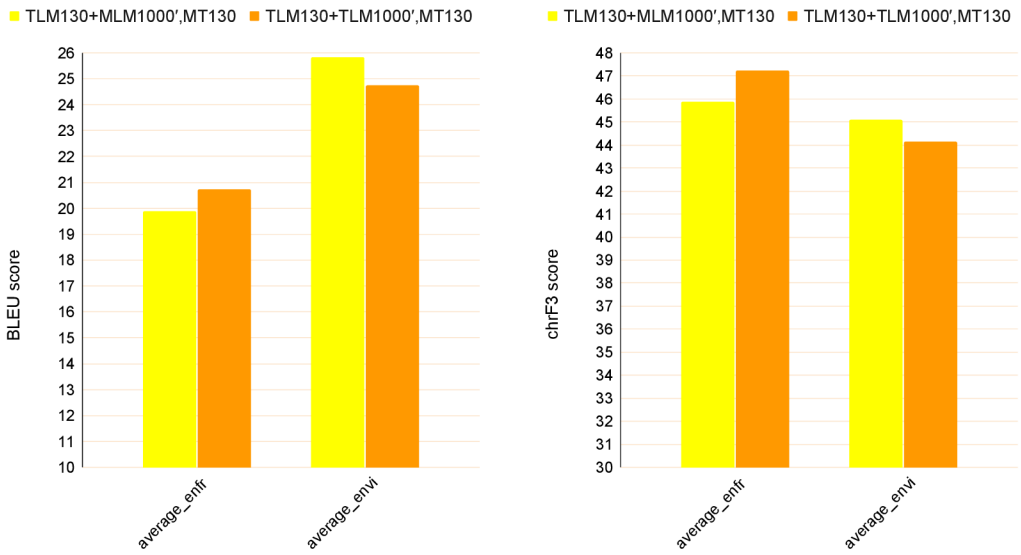
Displayed in the graph in Figure 5.7 (b), the evaluation of the performance of models described in Section 5.10 shows that the LMcat models outperform the LMsep models for the English-French language pair. At the same time, the former ones show inferior performance for the English-Vietnamese language pair.

The same, even though not as clear, can be said for the chrF3 score evaluation of LMcat models described in Section 5.7 for English-French and -Czech language pairs: they slightly outperform their counterparts. The opposite observation is also valid for the evaluation of the models for English-Vietnamese and -Basque pairs: LMsep models provide better results for each test set.

Both sets of experiments with different sizes of training sets, share the same properties that are relevant to this research: the coverage of words by bilingual dictionaries and the domain of data remained unchanged. The only difference was the amount of pseudo-parallel data. The increase in the size of which emphasized the tendency which was only barely observable with the models trained on small datasets: it appears that leveraging pseudo-parallel data with TLM objective is beneficial only for the target languages that are linguistically close to the source language (French, Czech) and harmful when the target language is distant from the source one (Vietnamese, Basque).



(a) Models trained on a limited amount of additional monolingual data



(b) Models trained on a large amount of additional monolingual data

Figure 5.7: Performance comparison of models that were treating the pseudo-parallel data differently: yellow – leveraging it through MLM objective (will be referred to as *LMsep*), orange – leveraging it through the TLM objective (will be referred to as *LMcat*).

5.12 Fine-tuning the XLM-R model

The last series of experiments and, perhaps, the one that is the closest to the real situation when it comes to training an MT model for low-resource language pair, resembles the case where there exists a large publicly available pre-trained model which is supposed to be fine-tuned for a concrete language pair for the MT task.

In order to fine-tune the XLM-R model, it is required to download the model itself, codes and vocabulary listed in the XLM repository and train the model in the way described before on data processed using downloaded codes and vocabulary.

5.12.1 XLM 17

For this set of experiments, *XLM 17* [37] is utilized as such a pre-trained model. XLM 17 is a multilingual model that was pre-trained with the MLM objective on corpora in 17 languages (en, fr, es, de, it, pt, nl, sv, pl, ru, ar, tr, zh, ja, ko, hi, vi). The model has 16 layers and 16 attention heads, it utilizes embedding vectors of size 1280 and a vocabulary size of 200k tokens.

5.12.2 Fine-tuning with MT objective

XLM 17 models fine-tuned with the MT objective suppose to serve as a Baseline. Tuning the model like this is the traditional way of adapting a language model for the MT task.

5.12.3 Two-stage fine-tuning

Two-stage fine-tuning of XLM 17 consists of first fine-tuning the model with TLM+MLM objectives on set that consists of the basic set and pseudo-parallel data and, second, of again fine-tuning it but with MT objective instead. It is expected that the performance of models for language pairs which have the target language covered during the XLM 17 training (French, Vietnamese) will be similar to the one of previously described MT fine-tuned models. It is so because the model will fail to optimize on provided data since the data was either already observed (true data from basic sets) or the data will have less quality (pseudo-parallel data) compared to the data provided for the initial XLM 17 LM training.

5.12.4 Results analysis

As is apparent from the performance graph from Figure 5.8, there is no visible improvement provided by the two-stage fine-tuning.

Like it was expected, the performance of MT-tuned models for language pairs with the target language covered by the initial XLM 17 training does differ from the performance provided by the models that were two-stage fine-tuned.

In addition to that 2-stage fine-tuned models also failed to provide performance improvement for 2 other language pairs. For the English-Czech language pair, it may be explained by the fact that the linguistic features that can be extracted from the provided corpora could have been already learned from the languages with similar linguistic features – XLM 17 was trained, among other languages, on Polish data. For English-Basque there are two possible explanations of such behaviour: first, similar to English-Czech, the linguistic features could have been learned from another agglutinative language with SOV (subject-object-verb) word order – XLM 17 was trained on Turkish data; second, shared sub-word vocabulary that is provided along with XLM 17 model is not optimized for the BPE encoding of the Basque language data and thus it is hard to learn the features specific to the Basque through the language modelling.

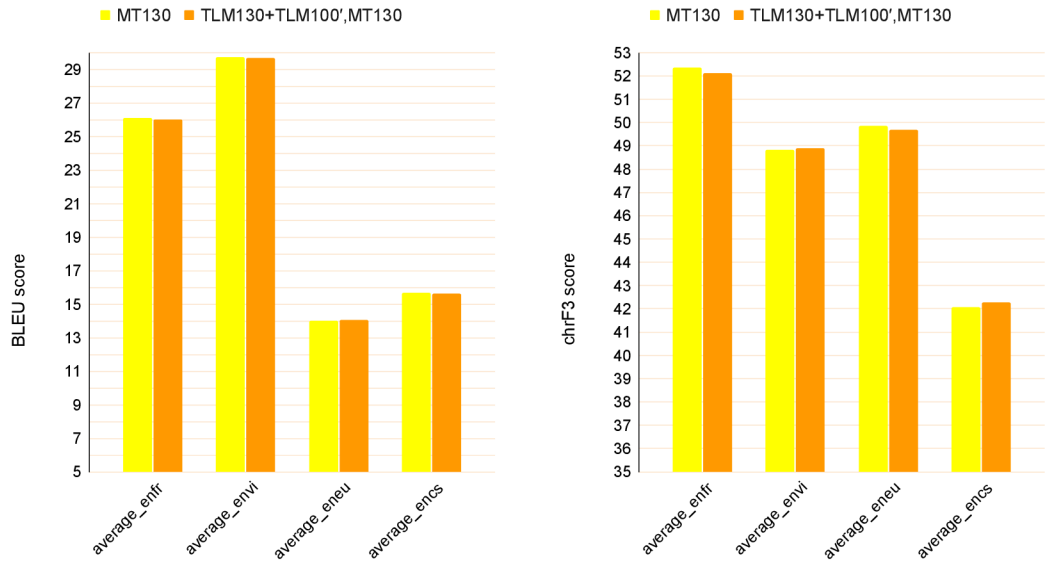


Figure 5.8: Performance comparison of MT fine-tuning/2-stage fine-tuning.

5.13 Findings

I will provide a brief overview of the findings of the BDBNMT derived from the conducted experiments.

- The higher bilingual dictionary coverage the bigger the yield the bilingual dictionary methods bring to the models' performance.
- Target languages with weak inflection (e.g. Vietnamese) get a bigger performance boost from the utilization of pseudo-parallel data.
- Leveraging the pseudo-parallel data through the TLM objective is justified only when the languages that make up a language pair are linguistically close (e.g. English-French).

Chapter 6

Future work

Even though the experiments conducted for this work demonstrate that the BDBNMT approach is applicable to many of the investigated cases, there exist a few additions to this approach that may positively affect the results which were not implemented and tested.

6.1 Lemmatization

It is obvious that the extension of bilingual dictionaries should increase the coverage of data and thus will be beneficial for the performance of models that adopt this approach. However, the induction of a ground-truth bilingual lexicon is not an easy task. Another way to improve the data coverage without any additions of ground-truth entries to the original dictionary is to lemmatize both the tokens in a monolingual corpus that were not covered by the initial dictionary and the keys of the bilingual dictionary itself. Consider the initial English-Czech dictionary {"fox": "liška", "lazy": "liný", "jumped": "přeskočil"} and the following pair of clean and noised versions of the same sentence:

Brown fox jumps over the lazy dog. → Brown liška jumps over the liný dog.

by the lemmatization of the initial dictionary we will obtain the following extension: {"jump": ["přeskočil", "přeskočit"]}, and after its application on the initial sentence we will obtain the sentence with the increased quantity of words translated by a bilingual dictionary:

Brown fox jump over the lazy dog. → Brown liška přeskočit over the liný dog.

Even though the form of the target word may not be correct, the greater bilingual dictionary coverage of the source monolingual dataset forces the model to drive the embeddings of similar words in different languages even closer during the pretraining.

6.2 Synonyms utilization

Another improvement that is even more reliant on a bilingual dictionary is that can be added in addition to previously described pseudo-parallel data generation and may be useful for extremely low-resource language pairs. This technique includes the usage of bilingual dictionaries in both directions: in opposition to the previously described process of generation of pseudo-parallel data we, instead of sampling the random ground-truth translation, now apply all of them by the creation of new target pseudo sentences for each available word translation. In addition to that we apply word mappings in the opposite

direction thus also enriching the source language data. Consider the following vocabulary {"country": ["země", "stát"], "land": "země", "ground": "země", "earth": "země",}. Since the word "has "country" has two corresponding words in the target language, by the application of substitutions we will generate two new target language pseudo sentences, then with the application of a dictionary in the opposite direction three new source sentences will be created that will cover the words in the source language that, perhaps, are not present in corpus. This way we will make the embedding vector representation of words involved in substitutions extremely close to each other (up to the point where embeddings of these words can be the same), which may be harmful to the performance of a model trained on a sufficient amount of data – even though the covered words may have the similar meaning they may be used in different contexts. However, as stated before, this technique may be useful for extremely low-resource languages, the corpus of which may not even include the words covered by a dictionary.

Chapter 7

Conclusion

The goal of this work was to investigate how NMT models' performance can be improved by the application of bilingual dictionary based methods. In this work were considered two methods of bilingual dictionary utilization that can be applied in certain situations.

The first one assumes the usage of *Anchored cross-lingual pretraining* (ACP) in conjunction with *Anchored training* (AT). Both methods expect the usage of non-parallel corpora for training. This research surveyed the setting that was not covered by the work which introduced these methods – the case when there are small-scale monolingual corpora. The research shows that there is no improvement that can ACP+AT training provide in this setting even when compared to the translations based solely on word mappings provided by the bilingual lexicon.

The second method – *Cross-lingual language modelling (XLM) with pseudo-parallel data* – assumes that there is a little amount of true parallel data available for Machine translation (MT) training and some amount of monolingual data available only in one of two languages. This technique consists of first generating the pseudo-parallel data based on a monolingual corpus and bilingual dictionary, and then leveraging this data through the language modelling objective. Conducted experiments proved that the addition of pseudo-parallel data to true parallel data is an effective method of boosting the model's performance: with the inclusion of a relatively small amount of extra monolingual data (100k source language sentences) it yielded improvement comparable to this achieved by the addition of true non-parallel data for both languages; with the bigger amount of additional monolingual data (1M source language sentences) it also led to performance improvement, however, it was more modest.

In addition to this survey, there was conducted the experiment that considers models that were trained with the ACP and AT stages as models for further fine-tuning with the MT objective. Pre-training with such objectives allows the model to also tune a decoder during the pre-training. The resulting performance of such models is comparable to this of the MT models pre-trained with the MLM objective on the same non-parallel data.

Another survey that was conducted for the method of XLM with the pseudo-parallel data was examining whether it is beneficial to leverage the pseudo-parallel data through the TLM objective. For the models trained in different settings, results show that it is advantageous to leverage the pseudo-parallel data this way when the linguistic distance between the source and target languages is relatively small – the improvement was observable for the models for English-French and English-Czech language pairs.

Bibliography

[1] Xiangyu Duan, Baijun Ji, Hao Jia, Min Tan, Min Zhang, Boxing Chen, Weihua Luo, and Yue Zhang. Bilingual dictionary based neural machine translation without using parallel sentences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1570–1579, Online, July 2020. Association for Computational Linguistics. 4, 3, 18, 23, 31, 35

[2] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederick Jelinek, John Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85, 1990. 3, 6

[3] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data, 2016. 3, 16

[4] Yasmin Moslem, Rejwanul Haque, John Kelleher, and Andy Way. Domain-specific text generation for machine translation. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 14–30, Orlando, USA, September 2022. Association for Machine Translation in the Americas. 3

[5] Yehoshua Bar-Hillel. A demonstration of the nonfeasibility of fully automatic high quality translation. 1960. 5

[6] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 2003. Association for Computational Linguistics. 7

[7] A. N. Jain, A. E. McNair, A. Waibel, H. Saito, A.G. Hauptmann, and J. Tebelskis. Connectionist and symbolic processing in speech-to-speech translation: The JANUS system. In *Proceedings of Machine Translation Summit III: Papers*, pages 113–117, Washington DC, USA, July 1-4 1991. 7

[8] Asunción Castaño, Francisco Casacuberta, and Enrique Vidal. Machine translation using neural networks and finite-state models. In *Proceedings of the 7th Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, St John's College, Santa Fe, July 23-25 1997. ?? Not mentionned on TOC. 7

[9] Holger Schwenk, Daniel Dechelotte, and Jean-Luc Gauvain. Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 723–730, Sydney, Australia, July 2006. Association for Computational Linguistics. 7

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. 7

[11] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. 7, 10

[12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. 7, 10

[13] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015. 7

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 7, 12

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 7, 17

[16] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 7

[17] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016. 7

[18] FIRTH J. R. A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*, 1957. 8

[19] A simple introduction to sequence to sequence models. https://www.analyticsvidhya.com/blog/2020/08/a-simple-introduction-to-sequence-to-sequence-models/. 9

[20] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014. 10

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. 14

[22] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. 15

[23] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining, 2019. 16, 17

[24] Jörg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). 23

[25] Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. Ccmatrix: Mining billions of high-quality parallel sentences on the web, 2019. 23

[26] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand, September 13-15 2005. 23

[27] Igor Azkarate. The web as a corpus of basque. 2014. 24

[28] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation, 2020. 24

[29] Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1856–1862, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). 24

[30] NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation. 2022. 25

[31] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019. 25

[32] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017. 25

[33] Mikel L. Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144, 2011. 25

[34] Jim O'Regan and Mikel L. Forcada. Peeking through the language barrier: the development of a free/open-source gisting system for basque to english based on apertium.org. *Procesamiento del Lenguaje Natural*, 51(0), 2013. 25

[35] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., 2009. 28

[36] Matt Post. A call for clarity in reporting bleu scores, 2018. 28

[37] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. 38

Appendix A

Code segments

A.1 Dictionary utilization

```
import pyonmttok
import random

tokenizer = pyonmttok.Tokenizer(mode='aggressive')
with open(src_path, 'r') as src, open(tgt_path, 'w') as tgt:
  sentences = []
    for line in src:
      tokens = tokenizer.tokenize(line)
      for idx, token in enumerate(tokens):
        if token in vocab:
          replaced += 1
          total += 1
        elif token.isalpha():
          total += 1
        if sample:
            token = random.sample(vocab.get(token, [token]), 1)[0]
        else:
            token = vocab.get(token, [token])[0]
      if replaced/float(total) >= 0.1:
        sentences.append(' '.join(tokens))
      else:
        sentences.append(line)
  tgt.write('\n'.join(sentences))
```

Algorithm A.1: Program segment responsible for the application of substitutions based on bilingual dictionary. First, it tokenizes the source sentence using the OpenNMT tokenizer's Python bindings[1], then apply the mappings provided by `vocab`. Each entry of *vocab* dictionary contains a list of translations read from the vocabulary file. Full algorithm that performs the reading of dictionary file and application of substitutions is provided by `sub_by_dict.py` script.

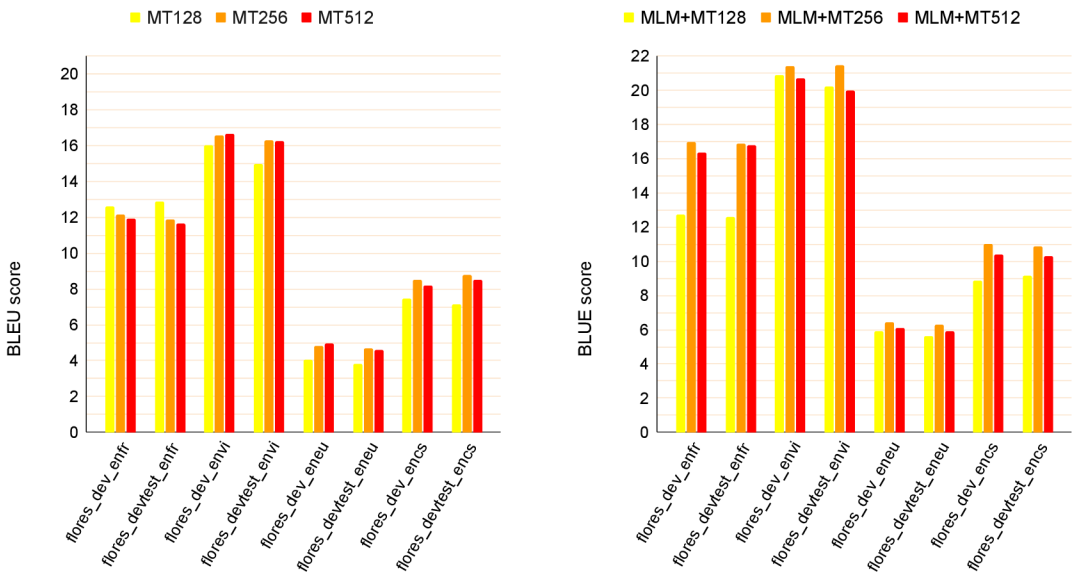
[1] https://github.com/OpenNMT/Tokenizer/tree/master/bindings/python

Appendix B

Determining the optimal embedding size

Given the fact that a huge parallel corpus (>1M sentences) is needed to train an NMT model and that the model will have only minimal data to use in the case of the first set of experiments described in Section 5.7; it was decided to optimise model performance by finding the optimal embedding size in advance for both pure MT training and MLM+MT training and then use the resulting optimal embedding size in subsequent experiments.

In order to do this, I tested the performance of models that were trained with 128, 256 and 512 embeddings sizes (other parameters remained unchanged for all 3 cases).



(a) Pure MT models

(b) Pre-trained MT models

Figure B.1: As it can be seen in graph (a) there is no definite superiority of models with some specific embedding size across language pairs. However, the case depicted in (b) – MT models that were first pre-trained with MLM objective – demonstrates that for both benchmarks and for all language pairs models that utilize embeddings of size 256 show a slight improvement in performance compared to those utilizing other embedding sizes.

Appendix C

Training arguments

In this appendix, I want to provide the reader with all the information needed to replicate the training experiments that were conducted during my research. Here will be listed 3 execution templates (for TLM/MLM pre-training, for BT training and MT training/fine-tuning) and tables corresponding to each of these templates, which will list the values of arguments to insert instead of corresponding placeholders.

```
python XLM/train.py  \
    --lgs en-<tgt> --mlm_steps <MLM steps> --emb_dim <embedding size> \
    --n_layers <number of layers> --n_heads <number of heads> \
    --dropout <dropout> --attention_dropout <attention dropout> \
    --gelu_activation true --batch_size <batch size> \
    --optimizer adam,lr=<learning rate>,weight_decay=<weight decay> \
    --epoch_size <size of training set> --max_epoch 1000 \
    --validation_metrics _valid_mlm_ppl --stopping_criterion _valid_mlm_ppl,10
```

Algorithm C.1: Template for the execution of LM training.

```
python XLM/train.py \
  --reload_model <model path> --lgs en-<tgt> --bt_steps en-<tgt>-en,<tgt>-en-<tgt> \
  --ae_steps en,<tgt> --lambda_ae 0:1,100000:0.1,300000:0 --encoder_only false \
  --word_dropout 0.1 --word_blank 0.1 --emb_dim <embedding size> \
  --n_layers <number of layers> --n_heads <number of heads> \
  --dropout <dropout> --attention_dropout <attention dropout> \
  --epoch_size <size of training set> --max_epoch 1000 \
  --gelu_activation true --batch_size <batch size> \
  --optimizer adam,lr=<learning rate>,weight_decay=<weight decay> \
  --eval_bleu true --eval_chrf true \
  --stopping_criterion valid_<language pair>_mt_chrf,10
```

Algorithm C.2: Template for the execution of BT training.

```
python XLM/train.py \
  --reload_model <model path> --lgs en-<tgt> --mt_steps en-<tgt> \
  --encoder_only false --emb_dim <embedding size> --n_layers <number of layers> \
  --n_heads <number of heads> --dropout <dropout> --attention_dropout <attention dropout> \
  --epoch_size <size of training set> --max_epoch 1000 \
  --gelu_activation true --batch_size <batch size> \
  --optimizer adam,lr=<learning rate>,weight_decay=<weight decay> \
  --eval_bleu true --eval_chrf true \
  --stopping_criterion valid_<language pair>_mt_chrf,10
```

Algorithm C.3: Template for the execution of MT training.

Hyperparameters enclosed within "<" and ">" that are listed in training execution templates are described in tables each corresponding to some experiment. If some parameter should be excluded, it is denoted as "-". The model path column corresponds to the path to the previously pre-trained model listed in the same group from one of the previous tables.

#	Model type	Model Path	MLM steps	E. size	N. of layers	N. of heads	Dropout	A. dropout	B. size	L. rate	W. decay	Codes
Models trained with a small amount of additional monolingual data.												
1	MLM	-	en,<tgt>	128	6	8	0.1	0.1	64	0.0001	0.1	5k
2		-		256								
3		-		512								
4	TLM	-	en,<tgt>,en-<tgt>	256								
Models pre-trained on a small amount of non-parallel data												
1	ACP	-	en,<tgt>	256	6	8	0.1	0.1	64	0.0001	0.1	5k
Models trained with a big amount of additional monolingual data.												
1	TLM	-	en,<tgt>,en-<tgt>	512	6	8	0.1	0.1	64	0.0001	0.1	7k
Fine-tuning the XLM 17 model.												
1	TLM	XLM17	en,<tgt>,en-<tgt>	1280	16	16	0.3	0.0	16	0.00001	0	175k

Table C.1: Arguments for LM training

#	Model type	Model path	E. size	N. of layers	N. of heads	Dropout	A. dropout	B. size	L. rate	W. decay	Codes
Models pre-trained on a small amount of non-parallel data											
1	ACP+AT	#1,#1	256	6	8	0.1	0.1	64	0.0001	0.1	5k

Table C.2: Arguments for BT (AT) training

#	Model type	Model path	E. size	N. of layers	N. of heads	Dropout	A. dropout	B. size	L. rate	W. decay	Codes
Models trained with a small amount of additional monolingual data											
1	Pure MT	-	128	6	8	0.1	0.1	64	0.0001	0.1	5k
2			256								
3			512								
4	MT FT	#1,#1	128								
5		#2,#2	256								
6		#3,#3	512								
7		#4,#4	256								
Models pre-trained on a small amount of non-parallel data											
1	ACP+AT,MT	#1,#1	256	6	8	0.1	0.1	64	0.0001	0.1	5k
Models trained with a big amount of additional monolingual data.											
1	MT FT	#1,#1	512	6	8	0.1	0.1	64	0.0001	0.1	7k
Fine-tuning the XLM 17 model.											
1	MT FT	XLM17,XLM17	1280	16	16	0.3	0.0	16	0.00001	0	175k
2	2-stage-FT	#1,#1									

Table C.3: Arguments for MT training

Appendix D

Models' performance overview

This appendix provides a table overview of the performance of all the models trained for this research on all test sets. Test sets denoted as *dev* and *test* correspond to flores200 dev and devtest sets respectively. BLEU and chrF3 scores are separated with a comma.

Model		en-fr			en-vi			en-eu			en-cs		
		dev	**test**	**Tatoeba**	**dev**	**test**	**Tatoeba**	**dev**	**test**	**Tatoeba**	**dev**	**test**	**Tatoeba**
Models trained with a small amount of additional monolingual data.													
MT	Emb128	12.58,	12.87,	15.24,	16.00,	14.95,	24.48,	4.06,	3.84,	17.9,	7.48,	7.12,	8.5,
	Emb256	12.16,39.4	11.86,39.42	13.25,36.27	16.54,36.41	16.29,35.56	22.94,40.6	4.8,37.69	4.68,37.14	16.24,45.59	8.5,34.25	8.79,34.62	8.64,29.7
	Emb512	11.9,	11.63,	13.45,	16.65,	16.25,	23.16,	4.96,	4.58,	16.3,	8.17,	8.5,	9.24,
MLM,MT	Emb128	12.74,	12.58,	14.52,	20.85,	20.22,	26.74,	5.91,	5.63,	20.19,	8.84,	9.15,	9.64,
	Emb256	16.96,43.96	16.87,44.39	17.87,40.71	21.39,41.22	21.43,41.06	25.5,43.53	6.42,40.33	6.28,39.93	18.45,48.24	10.99,37.72	10.88,38.06	12.43,33.01
	Emb512	16.33,	16.76,	16.98,	20.69,	19.97,	25.65,	6.12,	5.9,	17.73,	10.39,	10.31,	11.37,
TLM,MT	Baseline	18.67,46.15	18.19,46.24	19.89,42.71	22.44,42.18	22.39,42.07	26.71,44.56	6.79,41.17	6.9,41.01	19.85,49.56	11.44,38.7	11.17,38.45	11.48,32.84
	BDBsep	18.96,46.23	19.22,46.69	20.82,43.66	23.54,43.31	23.34,43.41	27.23,45.31	7.55,42.33	7.34,42.35	18.67,49.00	12.00,39.16	12.1,39.14	12.33,33.82
	BDBcat	19.3,47.04	18.91,47.03	19.53,43.17	22.24,42.34	22.99,42.83	26.77,44.84	7.05,41.92	6.81,41.73	18.71,49.07	11.74,39.39	11.73,39.71	12.08,34.18
	Topline	19.75,47.56	19.13,47.4	20.44,43.96	23.28,43.36	23.6,43.37	26.93,45.04	7.66,42.58	7.33,42.51	20.33,50.33	11.94,39.28	11.31,38.77	12.37,34.42
Models pre-trained on a small amount of non-parallel data.													
MLM,MT	Baseline	17.12,44.17	17.11,44.49	18.44,40.94	22.18,42.38	22.49,42.28	26.43,44.43	6.65,40.55	6.58,40.45	19.08,48.51	11.63,38.52	11.62,38.57	12.49,33.34
ACP,AT		2.70,25.77	2.50,25.51	2.37,19.20	3.21,18.10	2.95,17.68	2.45,14.08	1.67,21.34	1.63,21.26	1.28,17.3	3.27,22.05	3.18,21.93	1.39,13.48
ACP,AT,MT	Clean	19.90,47.18	19.18,47.20	19.69,42.17	22.67,42.4	22.32,42.21	25.79,43.64	7.42,41.80	6.99,41.89	18.34,48.30	11.6,38.60	11.08,38.83	11.73,32.88
	Noisy	7.36,33.25	7.25,33.46	6.34,26.39	12.12,32.12	12.08,31.81	11.04,29.13	4.25,33.41	4.08,33.28	10.33,35.03	3.80,25.49	3.80,25.42	3.50,17.58
Models trained with a big amount of additional monolingual data.													
TLM,MT	Baseline	18.66,46.25	19.49,47.31	20.23,44.43	22.94,43.06	23.15,43.02	26.99,45.49						
	Topline	20.46,48.32	20.87,48.90	23.34,46.48	24.96,44.82	25.07,44.85	28.72,46.85						
	TLMsep	18.97,46.34	18.92,46.74	21.76,44.59	24.77,44.62	24.58,44.55	28.13,46.13						
	TLMcat	20.07,47.81	20.48,48.21	21.6,45.65	23.53,43.66	23.33,43.23	27.35,45.57						
Fine-tuning the XLM 17 model.													
MT	Baseline	24.85,52.14	24.85,53.19	28.58,51.75	28.56,48.41	28.76,48.33	31.85,49.69	9.73,47.70	9.83,48.27	22.48,53.55	14.64,42.53	14.32,42.51	18.04,41.19
TLM,MT	TLMsep	24.05,51.73	25.08,52.56	28.85,52.11	28.63,48.37	28.83,48.45	31.6,49.85	10.00,47.48	9.50,47.69	22.72,53.92	14.30,42.16	14.57,42.69	17.98,41.96