Low-Resource Neural Machine Translation

Himanshu Arora
Mila, University of Montreal
himanshu.arora@umontreal.ca

Guillaume Lagrange
Mila, University of Montreal
guillaume.lagrange@umontreal.ca

Swetha Srikari Maganti
Mila, University of Montreal
swetha.srikari.maganti@umontreal.ca

Maziar Mohammad-Shahi
Mila, University of Montreal
maziar.mohammad-shahi@umontreal.ca

1 Introduction

The field of machine translation (MT), the automatic translation of source text from one natural language to another, has experienced a major paradigm shift in recent years. Previously dominated by statistical machine translation (SMT), which relies on various count-based models to map sentences from a source language to the target language, the field has largely been superseded by neural machine translation (NMT). NMT is an approach to machine translation that uses artificial neural networks to predict the likelihood of a sequence of words, often in the form of whole sentences. It has become the state-of-the-art for language pairs with large-scale parallel corpora. These advances, however, rely on the availability of large-scale parallel corpora to fit the hundreds of millions of model parameters required to make accurate predictions. On the other hand, the vast majority of languages today do not have such resources, meaning that the quality of machine translation for low-resource languages still leaves much to be desired. There are several approaches to overcome the lack of large parallel data, such as transfer learning, multi-task learning, semi-supervised and unsupervised learning techniques.

Here, we are interested in the task of neural machine translation from English to French in a low-resource scenario, where we only have access to 11,000 parallel examples. The translation system will need to generate French text with proper capitalization and punctuation using lower-cased English text without capitalization as input. Most MT systems however are trained on text data with proper punctuation, which makes the task at hand even more challenging. The input is similar to spoken language translation (SLT), an important part of automatic speech recognition (ASR) where the systems recognize sequences of words spoken, but do not provide punctuation marks (or capitalization). As with most low-resource languages, we can capitalize on larger unaligned monolingual corpora to help our NMT system: an English corpus of 474,000 examples and a French corpus of 474,000 examples.

In order to evaluate the quality of our translations, we compute the BLEU (bilingual evaluation understudy) score of the predicted outputs. The central idea behind BLEU is as follows: *the closer a machine translation is to a professional human translation, the better it is* [1]. To judge the quality of a machine translation, one measures its closeness to one or more reference human translations according to a numerical metric – the BLEU score. Scores are evaluated for individual translated segments by comparing them with the corresponding set of good quality human reference translations. Those scores are then averaged over the whole corpus to obtain an estimate of the overall translation quality. The approach works by counting matching n-grams in the candidate translation to n-grams in the reference text, where 1-gram or unigram would be each token and a bigram comparison would be each word pair. The comparison is made regardless of word order. Originally, BLEU's output was a number between 0 and 1, with values closer to 1 representing similar translations to the original references. In the results presented in the following sections, we follow the large body of recent NMT literature which represents the same BLEU score on a scale from 0 to 100 instead. More precisely, we follow the recommendations in [2].

In this paper, we explore different approaches to our machine translation problem using the small number of parallel examples as well as the larger monolingual corpora, and contribute a small Transformer architecture that performs significantly better than other alternatives explored with the help of synthetic data augmentation through back-translation and some clever pre/post-processing decisions. In Section 2, we provide an in-depth analysis of this

data. Section 3 reviews previous literature related to the subject at hand. Section 4 elaborates the methodology. A discussion of our experimental results is available in Section 5, before the paper concludes in Section 6.

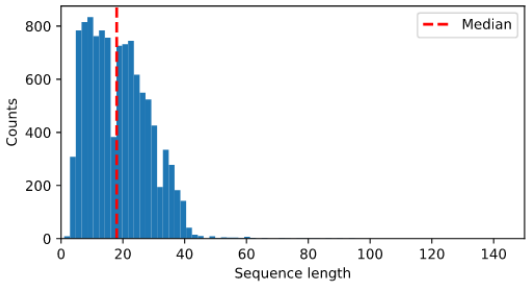
2 Data Analysis

As discussed in Section 1, the limited amount of parallel examples in English and French are characteristic of a low-resource NMT setting, with even the monolingual examples being limited when compared to common corpora usually leveraged for pre-training (e.g., BooksCorpus [4] with 800M words). As pointed out by Gu *et al.* [6], a NMT system cannot achieve reasonable translation quality when the number of parallel examples is extremely small ($N \approx$ 13k sentences in their analysis), which means that our scenario is actually an extremely low-resource setting, for which we have to try to leverage the monolingual data available. The statistics for the different English and French corpora are presented in Table 1.

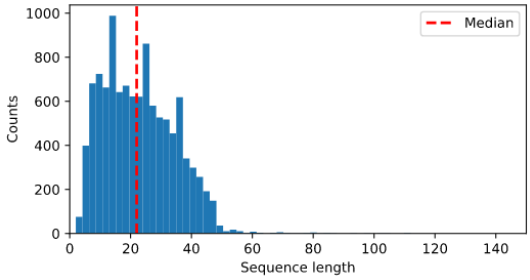
Table 1: Statistics for the parallel and monolingual sets used in the translation task. The number of words and the number of distinct words (case-insensitive) is based on the provided tokenizer (with the punctuation removed in both English sets).

	English Parallel	**French Parallel**	**English Monolingual**	**French Monolingual**
Examples	11,000	11,000	474,000	474,000
Words	205,361	251,176	8,820,344	10,764,893
Distinct words	13,657	17,360	60,008	83,154

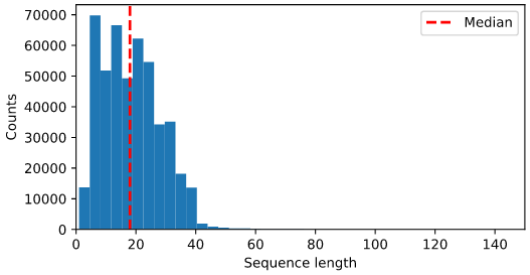
Both the parallel and monolingual corpora are made available to us in a text file format, with a single example per line. The text in both monolingual corpora is properly formatted, i.e. with proper capitalization and punctuation in both languages, and can be formatted like the parallel examples with the provided scripts. The parallel English examples are lower-cased and stripped of punctuation marks, while the French has proper capitalization and punctuation, making the task even more difficult. A first glimpse at the different corpora revealed similar sequence length distributions in the respective languages, as illustrated by Figure 1.



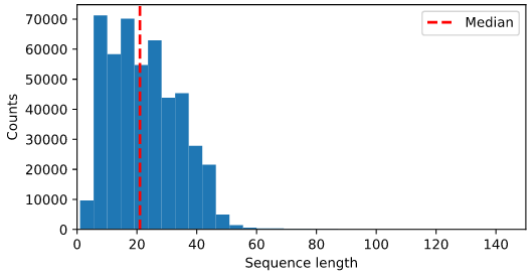
(a) English (parallel)



(b) French (parallel)



(c) English (monolingual)



(d) French (monolingual)

Figure 1: Sequence lengths distribution for the parallel and monolingual corpora.

To further characterize the nature of the data at hand, we wanted to analyze the topics discussed in the corpora. Topic modeling is a method for unsupervised classification of documents such as sentences, similar to clustering,

which finds natural groups of items. Latent Dirichlet allocation (LDA) is a particularly popular method for topic modeling. It treats each document as a mixture of topics, and each topic as a mixture of words. We pre-processed the sentences by removing common stopwords that are meaningless when fitting a topic model and performed lemmatization to increase the uniformity within the data. From the LDA analysis, we found that the most common topics were associated to "policy", "state", "european", "economic", "people", "president", "comission", "parliament", and "council". A similar conclusion was reached by doing simple part-of-speech (POS) tagging to count the frequency of the most common nouns, confirming that the parallel and monolingual corpora are related to European politics, perhaps even from the popular Europarl [3] dataset.

Interestingly, our analysis also exposed a number of bizarre samples that appear to be from a different topic distribution, with most of them also being shorter in sequence length. Although these appeared to be outliers in the different corpora distribution, they were not excluded from the examples since the blind test set was said to have a similar distribution, therefore we expect similar examples to also be present in the final test set. An example outlier can be found in *"go to school you lazy bastard"*, while the majority of European parliament related sequences sound like *"the measures proposed by the report in this regard are very courageous and interesting and they are very much in line with those being taken by the government of my country spain"*. More examples that clearly do not belong to the European parliament topics can be found in Appendix A. We also found an example where the source and target sentences actually contained a unit conversion from the imperial system to the metric system: 28 miles per hour in the English input corresponded to 50 km/h in the French target translation. Quite evidently, we do not expect our translation system to perform unit conversions.

3 Literature Review

Neural network architectures have become mainstream for machine translation in the past few years, taking over previous rule-based and phrase-based statistical MT models. The models proposed recently for neural machine translation often belong to a family of sequence-to-sequence (seq2seq) encoder-decoders and consist of an encoder, usually a recurrent neural network (RNN), that encodes a source sentence into a fixed-length vector from which a decoder (a second RNN) generates a translation. The key innovation, however, was the attention mechanism introduced by Bahdanau *et al.* [7], which allows the decoder to select at each step which part of the source sentence is more useful to consider for predicting the next word. The attention is a context vector that can be seen as modeling the alignment between input and output position.

For the first time at the 2016 Conference on Machine Translation (WMT16), a NMT system with attention-based encoder-decoder RNNs presented by Sennrich *et al.* [18] outperformed a phrase-based SMT on the news translation task. On top of the proposed model, the authors presented two additional improvements to reach their top performance: back-translation of monolingual target data to increase the amount of parallel data available for training [17], and the use of byte-pair encoding [16] to overcome the limited vocabulary of the encoder and decoder embeddings. Although the authors originally used a state-of-the-art phrase-based SMT engine to apply back-translation, back-translation (BT) has since been explored as an area of research on its own. It is a method that is simple and easy to apply as it does not require modification to the MT training algorithms, and an alternative to leverage monolingual data. Simply put, it requires training a target-to-source system in order to generate additional synthetic parallel data from the monolingual target data. This data complements the original parallel corpora to train the desired source-to-target system. There has also been work using source-side monolingual to improve translations, such as Zhang and Zong [25]. In their paper, the authors propose a self-learning and multi-task learning approach and observe that only source-side monolingual data related to the available parallel data helped in achieving better performance, while unrelated data made the performance worse. Furthermore, Hoand *et al.* [22] show how monolingual data from both languages can be leveraged by extending back-translation to dual learning or co-training. When training both source-to-target and target-to-source models jointly, one can use BT in both directions and perform multiple rounds of back-translation (namely, iterative back-translation) to improve the quality of the synthetic examples. A similar idea is applied by Lample *et al.* [14], where they also learn pre-trained word embeddings to initialize the encoder-decoder lookup table in their model and train their language model for each domain (source and target) before performing iterative back-translation. Similar to the first step of this approach, Qi *et al.* explore the potential of pre-trained word embeddings leveraging monolingual data for NMT, and show that they are most effective where there is very little training data but not so little that the system cannot be trained at all (performs very poorly).

While seq2seq encoder-decoder architectures have dominated recent NMT performances, they come at a large computational cost due to their use of RNNs. More recently, the Transformer model proposed by Vaswani *et al.* [8] removed sequential dependencies in the encoder and decoder networks, making use of self-attention networks and positional encoding instead. This resulted in much better GPU parallelization, thus substantially reducing the training time while outperforming previous RNN-based encoder-decoder models. Since then, the Transformer architecture

has been widely adopted by the research community in various natural language tasks like NMT, and has also seen a lot of attention for low-resource NMT. However, the popular Transformer configurations proposed by Vaswani *et al.* [8] usually do not transfer well on low-resource scenarios, where reducing the model size helps to overcome over-fitting. One way to do this is by tuning model hyper-parameters using different methods like standard grid search, or auto-sizing proposed by Murray *et al.* [21] which also used regularization. This method not only boosted the BLEU score but also resulted in faster training due to its reduced number of parameters.

More recently, pre-training techniques have without a doubt become one of the hottest research topics in Natural Language Processing (NLP), achieving great improvements when the data scale becomes large and the neural network models become bigger. The models are first pre-trained on large amount of unlabeled data to capture rich representations of the input, and then applied to downstream tasks by either providing context-aware embeddings of an input sequence or initializing the parameters of the downstream model for fine-tuning. Among them, BERT (Devlin *et al.* [9]) is one of the most prominent that inspired a lot of variants like XLM (Lample and Conneau [12]), MASS (Song *et al.*) [13]), XLNet (Yang *et al.* [10]) and RoBERTa (Liu *et al.* [11]) which achieve state-of-the-art performance for many natural language understanding tasks. Two kinds of objective functions are used in BERT training. The first is masked language modeling (MLM), where 15% of the words in a sentence are masked and BERT is trained to predict them with their surrounding words. The second is next sentence prediction (NSP), where the model learns to predict whether two input sequences are adjacent. This second task is removed from future variants like XLM, where the model is pre-trained based on multiple languages (cross-lingual). MASS adapts the ideas in BERT to generative tasks by jointly training both encoder and decoder in a seq2seq framework. In XLNet, masked language modeling is adapted to a permutation language modeling training objective, while in RoBERTa they use even more data and use a dynamic masking pattern.

Although these pre-training techniques show significant improvements on the downstream NMT task, and in some cases even in low-resource scenarios, they also make use of a much bigger monolingual corpora to pre-train their Transformer architecture. Datasets such as BooksCorpus (800M words) [4], English Wikipedia (2,500M words), English News Crawl (4,428M words) [5] and Common Crawl (65,128M words) [5] are often leveraged for the pre-training task (sometimes in combination), which represents a lot more data than what we have at hand (9.8M words for the English corpus).

4 Methodology

4.1 Data Processing & Pipeline

In neural machine translation (NMT), we are effectively dealing with text sequences and have to represent these sequences in an appropriate manner for our language model to understand. This common pre-processing step in natural language processing (NLP) is called tokenization, where sequences of text are broken into smaller parts, or tokens. A token can represent different unit types, with the most common approaches using word-level, subword-level or even character-level decomposition. In the experiments presented in Section 4.3, we explored word-level tokenization as a baseline, and then used a subword text encoder since translation is an open-vocabulary problem, and methods encoding rare and unknown words as sequences of subword units like byte-pair encoding (BPE) [16] have been proven to help in this task. The `SubwordTextEncoder`[1] works similarly to BPE, but does tokenization on spaces and punctuation jointly with splitting to subwords. Unlike BPE, it encodes also the spaces (or absence of spaces), so the raw sequence is fully reproducible when decoding the output. Inspired by `fastai`'s [29] tokenizer[2], we also replaced every capitalized word by its lower-case counterpart, adding a special `<maj>` token beforehand indicating that the first letter of the next word is capitalized, and every upper-cased word with its lower-case equivalent preceded by a special `<upp>` token for some of our experiments. This effectively limits the vocabulary size, while encoding capitalization in a way that is also easy to decode when post-processing.

In processing sequence data, it is also very common for individual samples to have different lengths. When scaling up a training model, it is also important to be able to feed in large volumes of mini-batches of variable-length data to take advantage of GPU parallelization. The most common method consists of zero padding all of the sequences to the longest sequence length, but it is also very inefficient. To overcome the inefficiency of this method, one can instead batch the sequences into the desired mini-batch size, and then pad them to the longest sequence length in this mini-batch, a method also known as dynamic padding. One can also go a step further, and instead perform a bucketing of the sequences, where each bucket is composed of sequences with similar lengths. Once bucketed, mini-batches are constructed and zero padded with minimal amount of padding since sequences in the same batch have similar

[1] `https://www.tensorflow.org/datasets/api_docs/python/tfds/features/text/SubwordTextEncoder`

[2] `https://docs.fast.ai/text.transform.html#Tokenizer`

lengths. We initially experimented with bucketing, but it can introduce a length-based bias during the training phase as it reduces the variability within batches. This is especially true during the validation step, where the predicted samples stop at the same length as the padded ground truths so as to reduce the evaluation time, but also to make the dimensions compatible with the targets when computing the validation loss (which we track, but do not report in our plots as mentioned in Section 4.3). Thus, we reverted to using dynamic padding for all of our experiments to reduce bias and have more intra-batch variability. The padding is ignored during training by masking the sequences properly.

4.2 Transformer Architecture

Based on the different configurations originally presented by Vaswani *et al.* [8], we explored smaller configurations as they have been proven to be better suited for low-resource settings, starting from the TRA_{BASE} variation presented in Table 2. The encoder and decoder are composed of a stack of N identical layers, where the first encoder takes positional information and embeddings of the input sequence as its input. The positional encodings have the same dimension d_{model} as the embeddings, so that the two can be summed, with dropout applied to the embeddings both the encoder and decoder stacks using a rate of P_{drop}. Each of the layers in the encoder and decoder contains a fully connected feed-forward network (Position-wise Feed-Forward Networks) with an inner-layer of dimensionality d_{ff}. The model also has h parallel attention layers (heads).

Table 2: Variations on the Transformer architecture.

Name	N	d_{model}	d_{ff}	h	P_{drop}
TRA_{XS}	2	128	512	8	0.1
TRA_{SM}	4	128	512	8	0.1
TRA_{MD}	4	256	1024	8	0.1
TRA_{BASE}	6	512	2048	8	0.1

4.3 Experiments

In order to have reproducible and comparable results, we set the random state (seed) to 128 for all of our experiments, with which the training set is always shuffled and a validation split of 10% is selected after shuffling (1,100 examples). The validation set is representative of the original corpus, with a median English sequence length of 18 words just like originally reported in Figure 1a. For the experiments where an augmented dataset is used, we first selected the same validation set so as to not introduce noisy examples to our targets, and then shuffled the original parallel training data (authentic) with the augmented data (synthetic). Although we are optimizing for a different metric (BLEU – which is not differentiable), we used the cross-entropy loss for all of our experiments (not for lack of differentiable alternatives to BLEU [24]). We used the Adam optimization algorithm (beta_1=0.9, beta_2=0.98, epsilon=1e-9) with a custom learning rate scheduler for all of our Transformer experiments according to the formula presented by Vaswani *et al.* [8]. All experiments were ran for up to 125 epochs based on the convergence of the models with a batch size of 128 (although it could be reduced in future works to explore its impact). We selected the best models through early stopping based on the BLEU score.

Sequence-to-sequence models are trained with teacher forcing [28]: instead of using the predicted output as the input at the next step, the ground truth targets are used. Without teacher forcing these models are much slower to converge, if they do so at all. On the other hand, this practice causes a mismatch between training and inference time. During training, the previous ground truth token is always known, but not during inference, where we use the previous prediction instead. Because of this difference, it is common to see a large gap between error rates on a held-out set evaluated with teacher forcing versus true inference. Furthermore, comparison of different language models may not be straightforward, especially if the models utilize different segmentation units (e.g., word-level versus subword units) as we are computing the *tokenwise* cross-entropies. Thus, because of the mismatch between training and inference caused by teacher forcing, and the fact that the loss is computed *tokenwise* instead of incorporating a sentence-level cost, we perform model selection based on the evaluation metric – the BLEU score. We followed this approach for all of our experiments, for which we report the training loss and validation BLEU score, which can be compared more easily across experiments. To follow the blind test evaluation metric, we averaged the sentence-level BLEU scores for the predicted hypotheses and report its value.

Based on the recent literature, we wanted to explore different settings for our low-resource machine translation task to establish the best approach. More precisely, we trained a Transformer model from scratch on the parallel data, using various architecture configurations and pre-processing methods as comparative baselines. Our attention was

focused on the Transformer architecture as it has been proven to outperform previous RNN-based approaches, even in low-resource settings, while being much faster to train[3].

To explore the impacts of punctuation for NMT, we trained a Bidirectional Long-Short Term Memory (BiLSTM) model based on the findings of Xu *et al.* [19] on the English monolingual corpora for the task of punctuation prediction (comma, period, exclamation mark and question mark). Given an input sequence of words, each word is labeled based on the punctuation before this word. The punctuation prediction only relies on lexical information, or simply word identities. We then used the trained model to infer punctuation on the English parallel sentences, which were used as source input instead of the original data stripped of punctuation. The implementation used[4] differs slightly from the original paper, as it also incorporates attention to the BiLSTM, and makes use of byte-pair encoding (BPE). We reproduced similar results to the original repository on our data, using a maximum sequence length of 96 (and thus discarding any longer sequence in the monolingual corpus). The experiment configurations and results can be found in Appendix D as they are not the focus of this report.

As pointed out by Qi *et al.* [20], pre-trained word embeddings have been found to be helpful in some low-resource scenarios for neural machine translation. For this reason, we also evaluated the impact of pre-trained contextual embeddings (fastText [15]) on the performance of our Transformer. For these experiments, we pre-trained source-side embeddings on the lower-cased English monolingual data stripped of punctuation, and target-side embeddings on the properly formatted French monolingual data. The embeddings were trained using the continuous bag-of-words (CBOW) algorithm for 30 epochs, with an embedding size $d_{\text{model}} = 128$. The other hyper-parameters were maintained to fastText's Python API defaults. The pre-trained embeddings for the English data were loaded on the encoder-side, while the ones for the French data were loaded on the decoder-side.

For our final experiments, we assessed the performance impact of growing the parallel through back-translation, as it has been proven by many authors to be helpful. For this, we first trained a target-to-source (FR→EN) NMT model with back-translation with the best configurations reported for the source-to-target model. The parallel training corpora was then expanded by inferring source sentences from the French monolingual corpora, which is used as additional targets with the corresponding noisy inputs that are the translated sentences. We preserve the original order of the samples when doing data augmentation with synthetic samples, so augmenting with 9,900 examples represents adding the first 9,900 French samples from the monolingual corpora and their corresponding translated samples before shuffling. As all papers reported that back-translated data are beneficial up to a certain point, we tried different authentic-to-synthetic data ratios as reported in Section 5. On the other hand, we also briefly explored self-learning as presented by Hoand *et al.* [25] – a semi-supervised method exploiting source-side monolingual data. For this, we used our baseline model to generate EN→FR translations and combined it with the parallel training corpora. This augmented data has 9,900 authentic examples and 8,960 synthetic examples in alternate batches to make it easier for freezing the decoder parameters with synthetic data while training since synthetic targets may negatively influence the decoder. Following He *et al.* [27], we applied dropout with $P_{\text{drop}} = 0.2$ instead since dropout plays an important role in boosting the BLEU score for self-training methods. We tried two different settings as reported in Section 5: first with 8,960 additional synthetic examples and then with 17,920 synthetic examples.

5 Results and Discussion

As mentioned in Section 4.3, we first experimented with different hyper-parameter configurations for our Transformer architecture. With this, we wanted to assess how each of them compared for our low-resource task using a simple word tokenizer, and it allowed us to set a good comparative baseline for our future experiments. Before fully leveraging the monolingual data available, we also wanted to evaluate the impact of capitalization in the target language, punctuation in the source language, and the segmentation of sequences into subword units with a smaller vocabulary. Table 3 shows the effect of adding different methods to the baseline NMT system, with the first 9 experiments not leveraging the monolingual data except for training the punctuation prediction BiLSTM model. In this section, we refer to an experiment by its associated (ID) in Table 3. The training loss and validation BLEU curves for all of our experiments can be found in Appendix C.

As demonstrated by our first experiments in Table 3, the bigger Transformer architectures actually decreased the performance, which is on par with the literature in low-resource scenarios, with TRA_{BASE} (4) achieving an extremely poor performance compared to the baseline. Regarding the impact of punctuation in the input language, even if it is

[3]We initially experimented with RNN-based methods like an encoder-decoder GRU with attention based on Tensorflow's tutorial, but noticed that they did not perform as well as our Transformer baseline. Thus, we chose to focus on the Transformer architecture in order to run the various experiments presented here that we thought would be more interesting to produce a cohesive report

[4]Implementation based on: `https://github.com/plkmo/NLP_Toolkit`

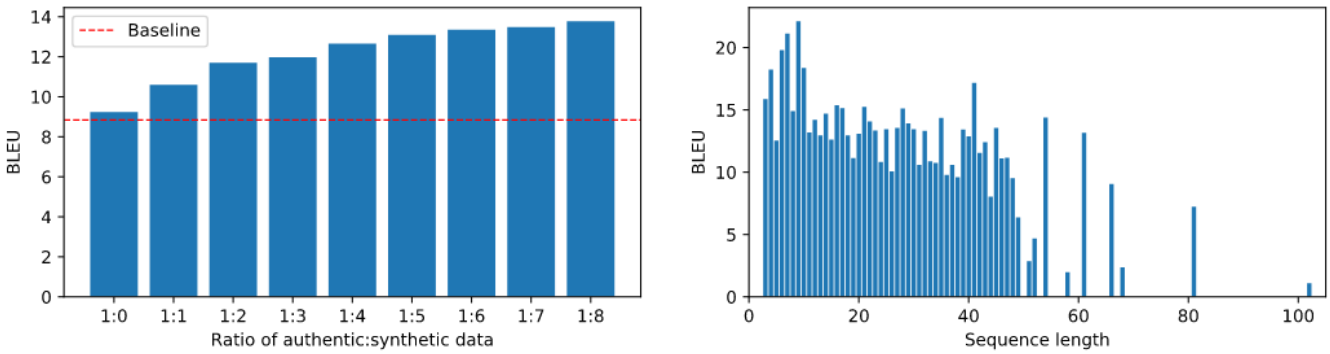
not always correctly predicted as demonstrated by the examples in Table 10 of Appendix D, it could still be said that the presence of punctuation in the source helps when we are trying to predict the properly formatted target translations since it increased the BLEU score on our validation set (6). On the other hand, reducing the target vocabulary size by lower-casing the words and adding special tokens to indicate capitalization (5) made the task easier, allowing our model to learn quite easily that the first letter of a sentence is capitalized, and words like "European", "Union", "Mr", "President" and "EU" should be capitalized in most contexts. Because inferring the punctuation on the data takes quite a lot more time to process, and that our BiLSTM is limited to a maximum sequence length of 96, we chose to follow-up using only capitalization encoding even though a combination of both approaches could help even more.

Table 3: Transformer (TRA) training experiments results on the validation set for English→French translation under different configurations.

ID	System	Early Stopping	BLEU
1	TRA_{XS} with word tokenization (baseline)	Epoch 110	8.84 (+0.00)
2	TRA_{SM} with word tokenization	Epoch 113	8.52 (-0.32)
3	TRA_{MD} with word tokenization	Epoch 124	8.43 (-0.41)
4	TRA_{BASE} with word tokenization	Epoch 23	3.54 (-5.30)
5	1 + lower-cased target with <maj> and <upp> tokens for capitalization	Epoch 124	8.99 (+0.15)
6	1 + punctuated input inferred from BiLSTM model	Epoch 66	8.95 (+0.11)
7	1 + subword segmentation (10k vocabulary)	Epoch 124	7.78 (-1.06)
8	7 + reduced subword vocabulary size (10k→2k)	Epoch 100	8.85 (+0.01)
9	8 + lower-cased target with <maj> and <upp> tokens for capitalization	Epoch 119	9.24 (+0.40)
10	1 + pre-trained fastText embeddings on the encoder side	Epoch 95	6.50 (-2.34)
11	10 + pre-trained fastText embeddings on the decoder side as well	Epoch 105	5.07 (-3.77)
12	9 + enlarged parallel corpora through back-translation (1:1 ratio)	Epoch 50	10.59 (+1.75)
13	9 + enlarged parallel corpora through back-translation (1:2 ratio)	Epoch 72	11.70 (+2.86)
14	9 + enlarged parallel corpora through back-translation (1:3 ratio)	Epoch 97	11.97 (+3.13)
15	9 + enlarged parallel corpora through back-translation (1:4 ratio)	Epoch 53	12.65 (+3.81)
16	9 + enlarged parallel corpora through back-translation (1:5 ratio)	Epoch 114	13.09 (+4.25)
17	9 + enlarged parallel corpora through back-translation (1:6 ratio)	Epoch 122	13.35 (+4.51)
18	9 + enlarged parallel corpora through back-translation (1:7 ratio)	Epoch 110	13.48 (+4.64)
19	9 + enlarged parallel corpora through back-translation (1:8 ratio)	Epoch 110	**13.77 (+4.93)**
20	5 + enlarged parallel corpora through self-learning (+8,960 samples)	Epoch 74	9.78 (+0.94)
21	5 + enlarged parallel corpora through self-learning (+17,920 samples)	Epoch 7	3.5 (-5.34)

As was found by Sennrich and Zhang [26], using a reduced subword vocabulary (8) increased the performance in a low-resource setting, although the performance is on-par with the baseline using simple word tokenization. Combined with capitalization encoding (9), we noticed a more significant improvement over the baseline ($+0.40$ BLEU). Contrary to Qi *et al.* [20], pre-training the word embeddings in the source (10) and target languages (11) did not help to increase the BLEU scores, and actually negatively impacted our system in both cases. In their findings, the authors state that the gain is highest when the baseline system is poor but not too poor, usually with a baseline BLEU score in the range of 3-4. This is not the case with our baseline, which scored 8.84 BLEU. Since our baseline was not as poor and already produced reasonable translations, it is probably why the pre-trained embeddings were not useful. The authors also used word embeddings of bigger dimension (300) pre-trained on a much larger corpus that is Wikipedia dumps, which could also explain the difference in results.

On the other hand, the results of our back-translation experiments were on par with the literature. As demonstrated by the results of our back-translation experiments (12-19), and more explicitly illustrated in Figure 2a, augmenting the training data with synthetic data generated through back-translation helped greatly. The target-to-source (FR→EN) system configuration and experiment results can be found in Appendix B. We noticed that the most significant gains arose from doubling (ratio 1:1) and tripling (ratio 1:2) the original training dataset size, while adding even more data only brings diminishing returns. Despite small incremental improvements on the validation BLEU score, bigger authentic-to-synthetic data ratios came with the price of a prolonged training time. In some papers, the authors mentioned using a ratio upwards of 1:10, at which point synthetic data clearly becomes harmful depending on the set-up as the score stops improving. Here, we did not manage to reach such conclusions based on our experiments, but we clearly see how the improvements are diminishing, and how more synthetic data might become harmful to our



(a) BLEU scores w.r.t. authentic-to-synthetic training data ratios.

(b) Average translation BLEU scores w.r.t. target sequence length.

Figure 2: A comparison of BLEU scores for our final back-translation experiments, where the 1:0 ratio in (a) represents the result of experiment 9.

system. It is still interesting to see how, even with noisy inputs, the system is able to learn better representations which are helpful to the translation quality. From the results of self-learning experiments (20-21), we noticed that using source-side monolingual data through self-learning did not improve the baseline substantially. Unlike back-translation, using more synthetic data actually degraded the performance of the system. The authors originally stated that more unrelated monolingual data reduces the performance [25], but our monolingual data is supposed to be from the same distribution. Thus, perhaps our results are not as promising simply because learning representations for the decoder is more difficult than the encoder, and so back-translation is a more suited approach to our current problem. However, these conclusions cannot be entirely confirmed based on 2 experiments.

Table 4: An example of when different symbols for punctuation marks hinder the BLEU score.

Input	**Target**	**Predicted**	**BLEU**
energy is the fundamental issue	L’ énergie est la question fondamentale .	L' énergie est la question fondamentale .	80.91

Analyzing the predicted translations of our best model, we observed some interesting phenomena. First of all, we noticed that, on average, longer target sequences are usually more difficult to predict correctly. As illustrated in Figure 2b, the average BLEU score seems to generally decrease with the target sequence length. We also noticed that the different symbols used for the same punctuation marks in the target language have an impact on the performance of our system because of the way the BLEU score is computed (e.g., the apostrophe ' or ’). For example, the predicted translation in Table 4 would actually correspond to a BLEU score of 100. Additionally, there is an inherent, systemic problem with any metric based on comparing with a single reference translation. In real life, sentences can be translated in many different ways, sometimes with no overlap. For our evaluation, we only have access to one reference translation, which is not a real indicator of the translation quality. Some examples of this phenomenon are depicted in Table 5, where most the the predicted translations are actually correct or partially correct, but have a much lower BLEU score because we only have one reference. In some cases, we also noticed that numerical values in the the input sometimes had their word equivalent in the target reference (e.g., 10→dix), but the system predicted the numerical value, for which it was penalized on the BLEU score.

In Appendix E, we visualize the attention patterns produced by the last attention heads of our Transformer on a sample source-target translation. With this animated visualization, we demonstrate how the model is translating the input sequence.

Table 5: Examples of predicted translations that are actually correct (or mostly correct), but have a lower than expected BLEU score because we only have one true reference.

Input	Target	Predicted	BLEU
this work has already begun	Ce travail commence dès maintenant .	Ce travail a déjà commencé .	17.97
switzerland was mentioned	On a parlé de la Suisse .	La Suède a été mentionnée .	8.17
cambridge – will military power become less important in the coming decades	CAMBRIDGE – Le pouvoir militaire perdra - t - il de son ascendance dans les années à venir ?	CAMBRIDGE – Le pouvoir militaire est devenu le moins important dans les décennies de décennies .	22.57
success is possible	On peut réussir .	Le succès est possible .	10.68
this is precisely why we must focus on the following objectives	Voilà exactement pourquoi nous devons nous concentrer sur les objectifs suivants .	C' est précisément pourquoi nous devons concentrer sur les objectifs suivants .	53.11
the european union must be able to intervene in order to maintain peace and security both inside and outside its borders	L' engagement démocratique du Parlement européen au sein des deuxième et troisième piliers doit être renforcé .	L' Union européenne doit pouvoir faire preuve de la stabilité afin de maintenir la paix et la sécurité des frontières et des frontières extérieures .	2.25

6 Conclusion

After performing a lot of experiments using different sizes of the Transformer architecture, segmentation units, pre/post-processing techniques and data augmentation techniques, we conclude that our proposed Transformer (TRA_{XS}) with subword segmentation and target-side capitalization encoding using an enlarged parallel corpora through back-translation performs significantly better than our baseline translation system. Given the points discussed in Section 5 related to the limitations of the BLEU evaluation metric under our circumstances, we expect our model to generalize well and perform similarly on the test set (with perhaps a slightly lower BLEU score due to the smaller number of examples in our validation set).

The work presented was limited by the time and resources available. Additionally, there are a lot of other methods that have not been explored in our work that could enhance the performance of our approach. Perhaps the popular pre-processing step of truecasing the input, i.e. determining the proper capitalization of words, would help since the target language is capitalized. One could also explore different decoding methods like *beam search*, *top-K sampling* or *top-p sampling* instead of the standard greedy search approach since it has been proven that these can help produce more natural output, and thus improve translation quality. To further improve the quality of our synthetic samples, one could also explore iterative back-translation. It would also be interesting to see how the batch size affects the performance of our system in a low-resource scenario, as it has been proven, in some cases, to improve the final BLEU score (although by a small margin).

As there are so many different approaches in the recent literature that are promising for NMT, unsupervised NMT and low-resource scenarios, it would be difficult to realistically explore them all. Nonetheless, it would be interesting to see how recent advances in pre-training methods such as BERT, XLM or MASS compare in a much lower resource setting like ours, where we only have 474,000 monolingual sentences (9.8M words) to leverage for pre-training before fine-tuning on our downstream task.

References

[1] Papinesi, K., Roukos, S., Ward, T. & Zhu, W-J (2002). *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. 311–318.

[2] Post, M. (2018). *A Call for Clarity in Reporting BLEU Scores*. (arXiv: 1804.08771)

[3] Koehn, P. (2005). *Europarl: A Parallel Corpus for Statistical Machine Translation*. Latest version of the corpus at `https://www.statmt.org/europarl/`.

[4] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A. & Fidler, S. (2015). *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*. (arXiv:1506.06724)

[5] Bojar, O., Chatterjee, R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Yepes, A.J., Koehn, P. & Monz, C. (2018). *Findings of the 2018 Conference on Machine Translation (WMT18)*. Proceedings of the Third Conference on Machine Translation (WMT), Brussels, pp. 272–303.

[6] Gu, J., Hassan, H., Devlin, J. & Li, V.O.K. (2018). *Universal Neural Machine Translation for Extremely Low Resource Languages*. (arXiv:1802.05368)

[7] Bahdanau, D., Cho, K., & Bengio, Y. (2014) *Neural Machine Translation by Jointly Learning to Align and Translate*. (arXiv: 1409.0473)

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. & Polosukhin, I. (2017). *Attention Is All You Need*. (arXiv:1706.03762)

[9] Devlin, J., Chang, M-W., Lee, K. & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. (arXiv:1810.04805)

[10] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. & Le, Q.V. (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. (arXiv:1906.08237)

[11] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. (arXiv:1907.11692)

[12] Lample, G. & Conneau, A. (2019). *Cross-lingual Language Model Pretraining*. (arXiv:1901.07291)

[13] Song, K., Tan, X., Qin, T., Lu, J. & Liu, T-Y. (2019). *MASS: Masked Sequence to Sequence Pre-training for Language Generation*. (arXiv: 1905.02450)

[14] Lample, G., Ott, M., Conneau, A., Denoyer, L. & Ranzato, M.A. (2018). *Phrase-Based & Neural Unsupervised Machine Translation*. (arXiv:1804.07755)

[15] Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2016). *Enriching Word Vectors with Subword Information*. (arXiv:1607.04606)

[16] Sennrich, R., Haddow, B. & Birch, A. (2015). *Neural Machine Translation of Rare Words with Subword Units*. (arXiv:1508.07909)

[17] Sennrich, R., Haddow, B. & Birch, A. (2015). *Improving Neural Machine Translation Models with Monolingual Data*. (arXiv:1511.06709)

[18] Sennrich, R., Haddow, B. & Birch, A. (2016). *Edinburgh Neural Machine Translation Systems for WMT 16*. (arXiv:1606.02891)

[19] Xu, K., Xie, L. & Yao, K. (2016). *Investigating LSTM for Punctuation Prediction*. Proceedings of the 10th International Symposium on Chinese Spoken Language Processing (ISCSLP).

[20] Qi, Y., Sachan, D.S, Felix, M., Padmanabhan, S.J. & Neubig, G. *When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation*. (arXiv:1804.06323)

[21] Murray, K., Kinnison, J., Nguyen, T.Q., Scheirer, W. & Chiang, D. (2019). *Auto-Sizing the Transformer Network: Improving Speed, Efficiency, and Performance for Low-Resource Machine Translation*. (arXiv:1910.06717)

[22] Hoand, C.D.V, Koehn, P., Haffari, G. & Cohn, T. (2018). *Iterative Back-Translation for Neural Machine Translation*. Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, Melbourne, pp. 18–24.

[23] Przystupa, M. & Abdul-Mageed, M. (2019). *Neural Machine Translation of Low-Resource and Similar Languages with Backtranslation*. Proceedings of the Fourth Conference on Machine Translation (WMT), Florence, pp. 224–235.

[24] Zhukov, V., Golikov, E. & Kretov, M. (2017). *Differentiable lower bound for expected BLEU score*. (arXiv:1712.04708)

[25] Zhang, J. & Zong, C. (2016). *Exploiting Source-side Monolingual Data in Neural Machine Translation*. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545.

[26] Sennrich, R. & Zhang, B. (2019). *Revisiting Low-Resource Neural Machine Translation: A Case Study*. (arXiv:1905.11901)

[27] He, J., Gu, J., Shen, J. & Ranzato, M.A. (2019). *Revisiting self-training for neural sequence generation*. (arXiv:1909.13788)

[28] Williams, R.J. & Zipser, D. (1989). *A learning algorithm for continually running fully recurrent neural networks*. Neural computation, **1**(2), pp. 270–280.

[29] Howard, J. & Gugger, S. (2020). *fastai: A Layered API for Deep Learning*. (arXiv:2002.04688)

Appendix

A Examples of Natural Topic Outliers Found in the Parallel Corpus

Table 6: Examples of natural topic outliers found in the parallel corpus.

Example	Words
you grabbed my ass	4
the girl buys milk at the market	7
robert was so busy he has to turn down an invitation to play golf	14
go to school you lazy bastard	6
then came bollywood	3
he wore old shoes	4
this teacher goes by the name of pops	8
pardon me but that is my racket	7

B FR→EN Experiment & Configuration

Table 7: Transformer (TRA) training experiment result on the validation set for French→English translation, using lower-cased French examples stripped of punctuation to make the back-translation task easier since the English parallel examples are not properly formatted.

ID	System	Early Stopping	BLEU
1	TRA_{XS} with subword segmentation (2k vocabulary)	Epoch 93	10.07

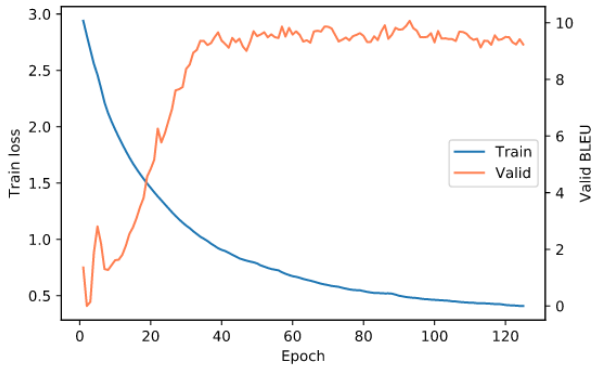
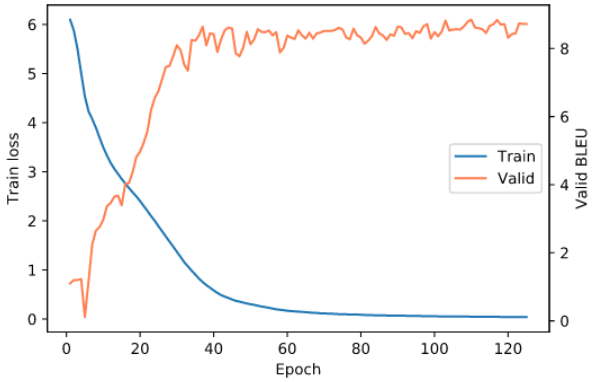
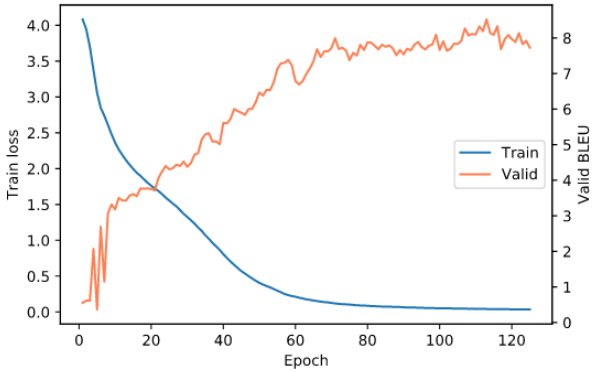


Figure 3: Train loss and validation BLEU for our FR→EN translation system used to generate back-translated samples.

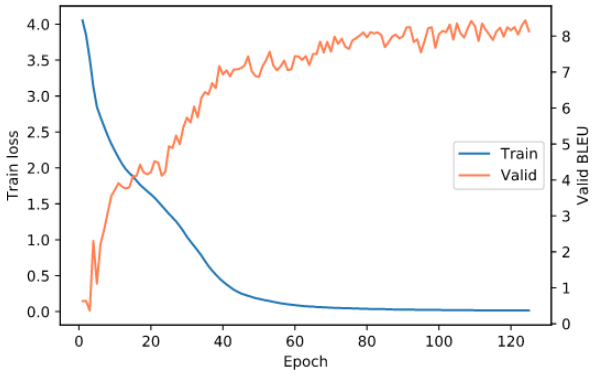
C EN→FR Experiments



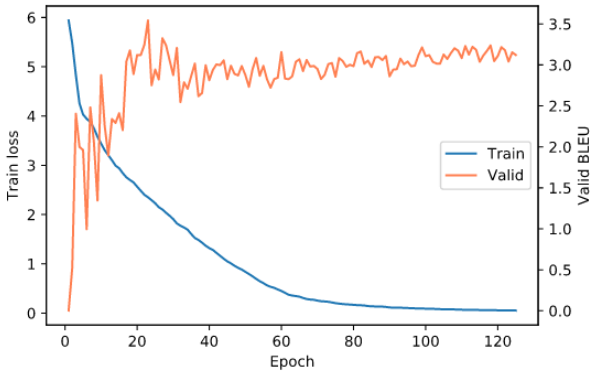
(a) TRA_{XS} with word tokenization.



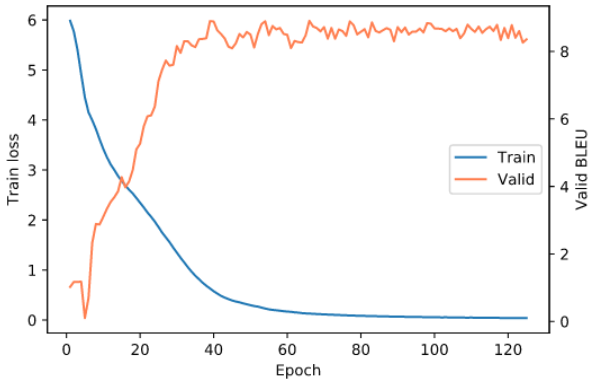
(b) TRA_{SM} with word tokenization.



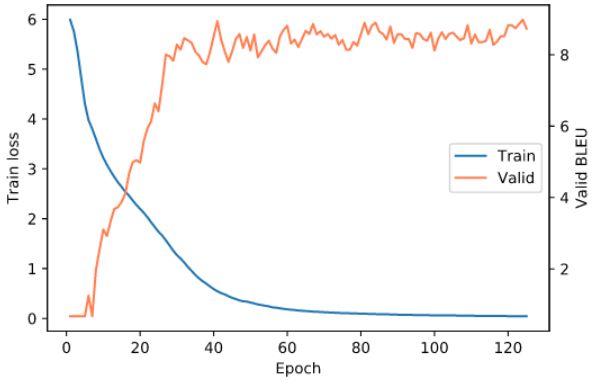
(c) TRA_{MD} with word tokenization.



(d) TRA_{BASE} with word tokenization.

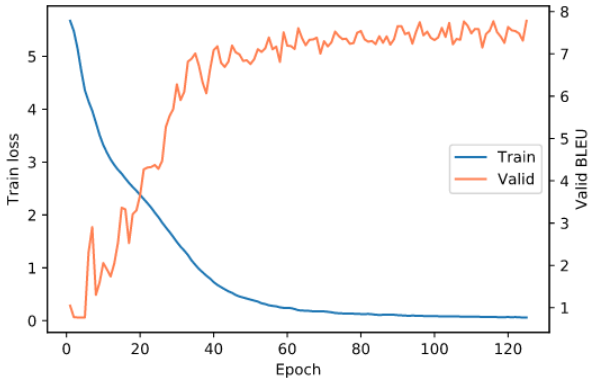


(e) TRA_{XS} with word tokenization on punctuated input.

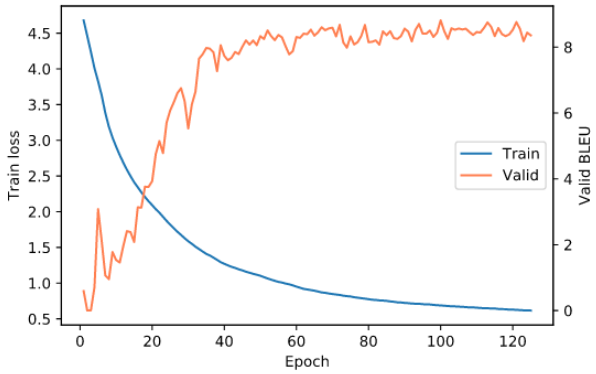


(f) TRA_{XS} with lower-cased target using `<maj>` and `<upp>` tokens for word capitalization.

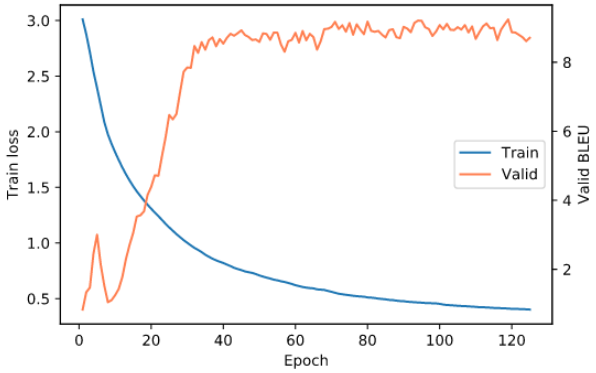
Figure 4: Train loss and validation BLEU for our different EN→FR experiments with word tokenization.



(a) TRA$_{XS}$ with subword tokenization (10k vocabulary).

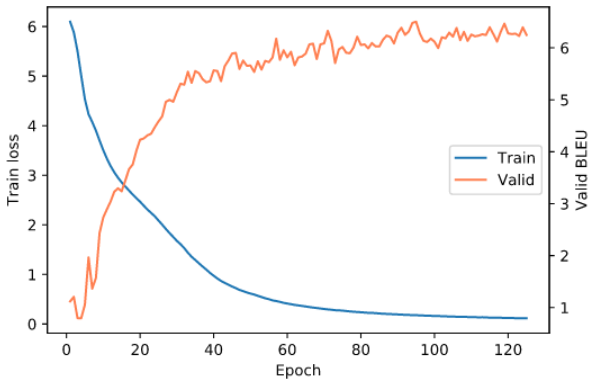


(b) TRA$_{XS}$ with subword tokenization (2k vocabulary).

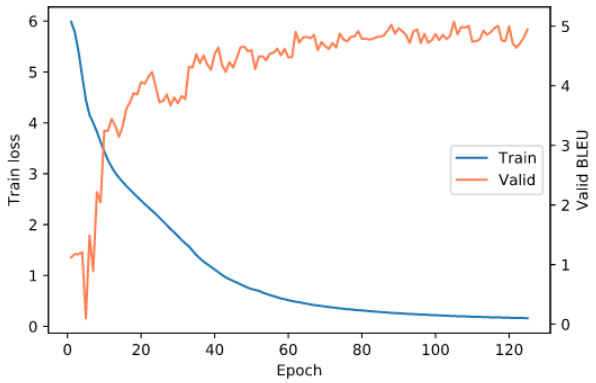


(c) TRA$_{XS}$ with subword tokenization (2k vocabulary) and lowercased target using `<maj>` and `<upp>` tokens for word capitalization.

Figure 5: Train loss and validation BLEU for our different EN→FR experiments with subword tokenization.

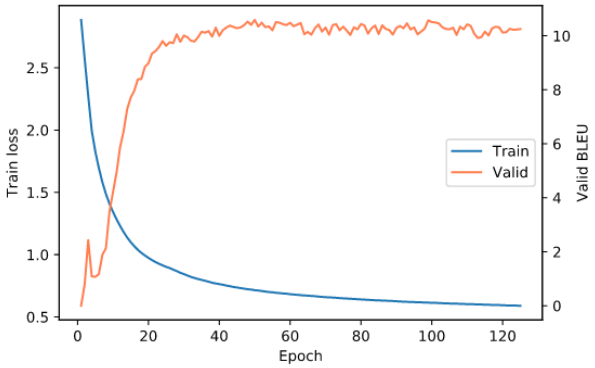


(a) TRA$_{XS}$ with word tokenization, using pre-trained embeddings for the encoder (English).

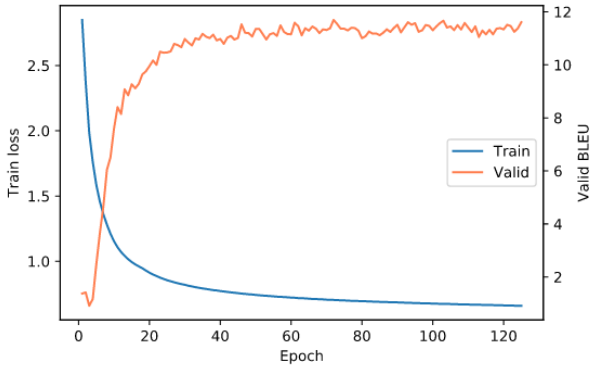


(b) TRA$_{XS}$ with word tokenization, using pre-trained embeddings for the encoder (English) and decoder (French).

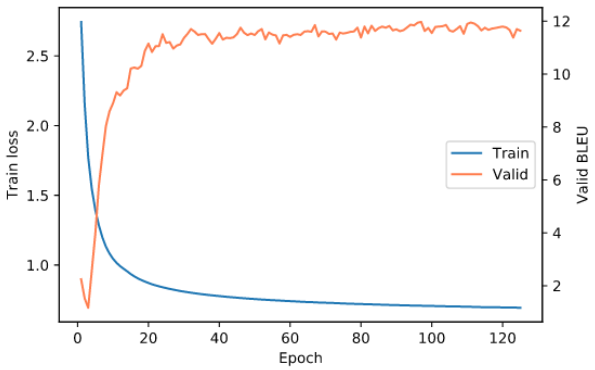
Figure 6: Train loss and validation BLEU for our different EN→FR experiments using fastText embeddings pre-trained on the monolingual corpora.



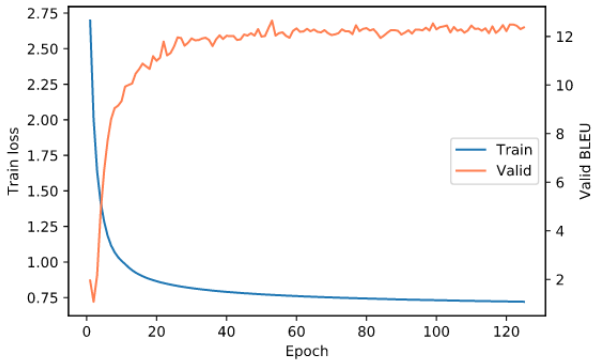
(a) Augmented training data (1:1 authentic-to-synthetic ratio).



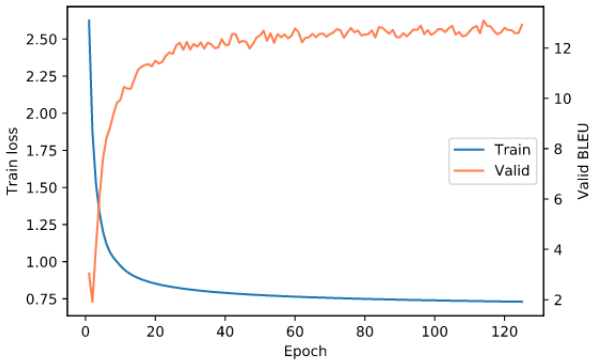
(b) Augmented training data (1:2 authentic-to-synthetic ratio).



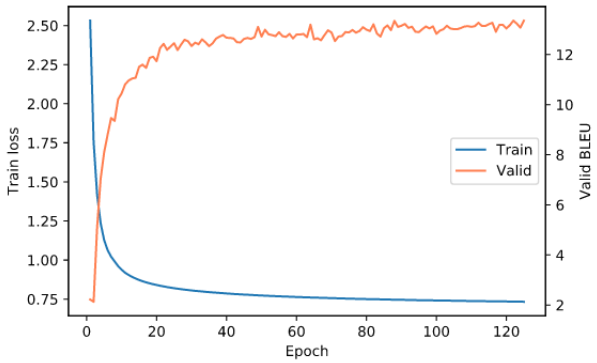
(c) Augmented training data (1:3 authentic-to-synthetic ratio).



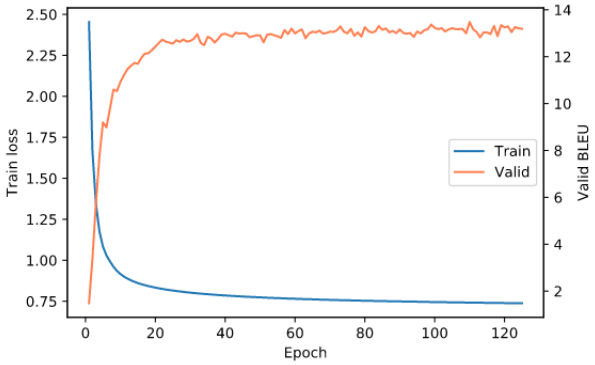
(d) Augmented training data (1:4 authentic-to-synthetic ratio).



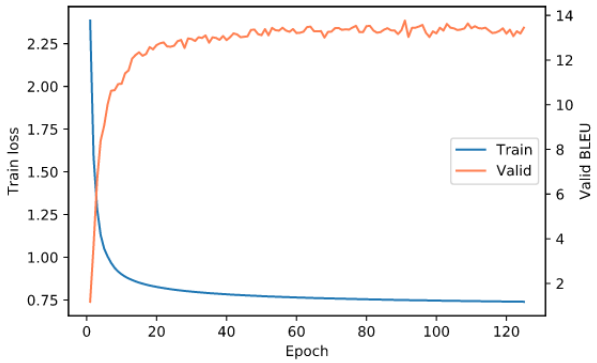
(e) Augmented training data (1:5 authentic-to-synthetic ratio).



(f) Augmented training data (1:6 authentic-to-synthetic ratio).

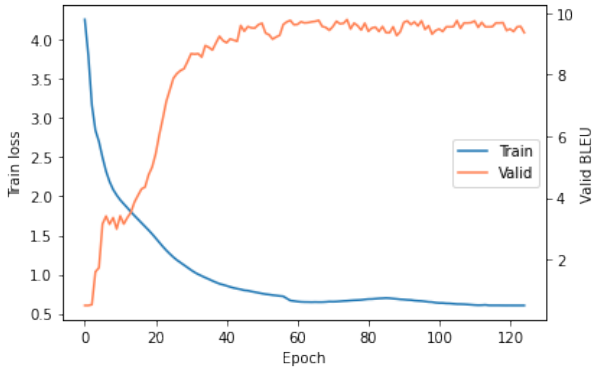


(g) Augmented training data (1:7 authentic-to-synthetic ratio).

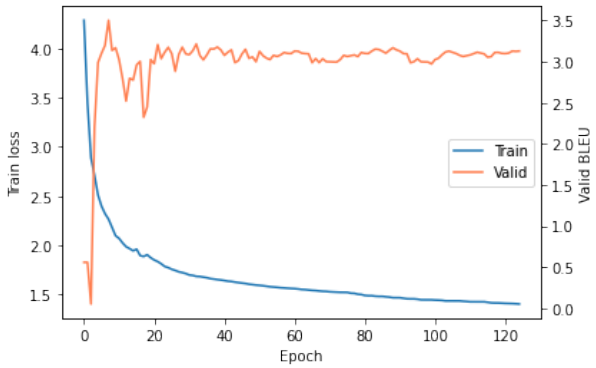


(h) Augmented training data (1:8 authentic-to-synthetic ratio).

Figure 7: Train loss and validation BLEU for our different EN→FR experiments using an augmented training corpus through back-translation. All experiments use TRA_{XS} with subword tokenization (2k vocabulary) and lower-cased target using `<maj>` and `<upp>` tokens for word capitalization.



(a) Augmented training data with 8960 synthetic examples



(b) Augmented training data with 17920 synthetic examples

Figure 8: Train loss and validation BLEU for our different EN→FR experiments using an augmented training corpus through self-learning. All experiments use TRA_{XS} with 0.2 dropout, word tokenization and lower-cased target using `<maj>` and `<upp>` tokens for word capitalization.

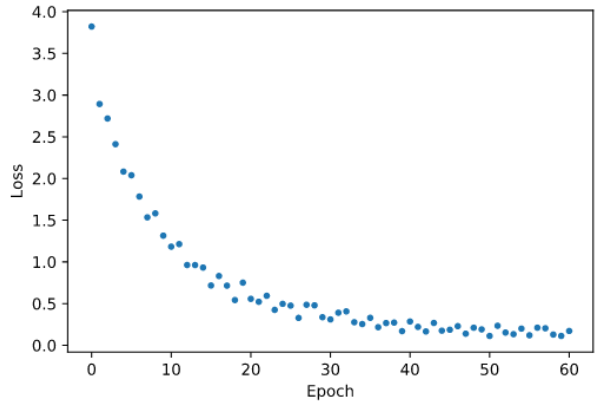
D Punctuation Prediction

Table 8: BiLSTM punctuation prediction experiment configurations.

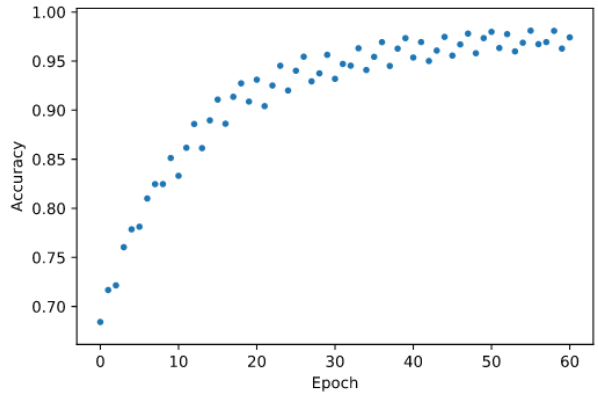
BPE Vocab	**Sequence Length**	**Embeddings Size**	**Hidden Size**	**Epochs**	**Learning Rate**
4085	96	512	512	65	0.0005

Table 9: BiLSTM punctuation prediction performance metrics on the English monolingual training data.

Accuracy (%)	**Precision** (%)	**Recall** (%)	**F1 Score** (%)
97.41	83.11	72.49	77.44



(a) Cross-entropy loss.



(b) Accuracy of predicted punctuation marks.

Figure 9: Train loss and accuracy of the BiLSTM punctuation prediction experiment.

Table 10: BiLSTM punctuation prediction examples, good and bad. It is also important to note that the tokenizer used in the repository removes some alphanumeric symbols like parentheses or hyphens, which could be slightly problematic.

Original	**Punctuated Prediction**
mr president ladies and gentlemen the financial perspective outlines the scope of the eu 's activities over coming years as well as providing a framework for such activities and determining how effective they will be	mr president , ladies and gentlemen , the financial perspective outlines the scope of the eu 's activities over coming years as well as providing a framework for such activities and determining how effective they will be
in addition it is vital to use energy efficiency	in addition , it is vital to use energy efficiency .
therefore my group will not support these amendments	therefore , my group will not support these amendments .
i agree with you that yes the member states are responsible but there are certain rules	i agree with you that yes , the member states are responsible , but there are certain rules .
what can europe do	what can europe do ?
if the passengers were growing restive they would rise	if the passengers were growing restive , they would rise .
however the general procedures for interoperability must not be defined by a hard case	however , the general procedures for interoperability must not be defined by a hard case .
in our enthusiasm we said that this had to be within three weeks	in our enthusiasm , we said that this had to be within three weeks .
you send one message and it goes out to everybody and you receive the messages you 're interested in	you send one message and it goes out to ? everybody and you receive the messages you 're interested in
as regards the scope of this security policy there is deliberate ambiguity	as regards the scope of this security , , policy there is deliberate ambiguity .
my group is therefore four - square behind the proposal in order for example to be able to coordinate rural development much better together under the umbrella of cohesion policy than in the past	my group is therefore four square behind , the proposal , in order for example to be able to coordinate rural development much better , together under the umbrella of cohesion policy , than in the past .

E Attention Visualization

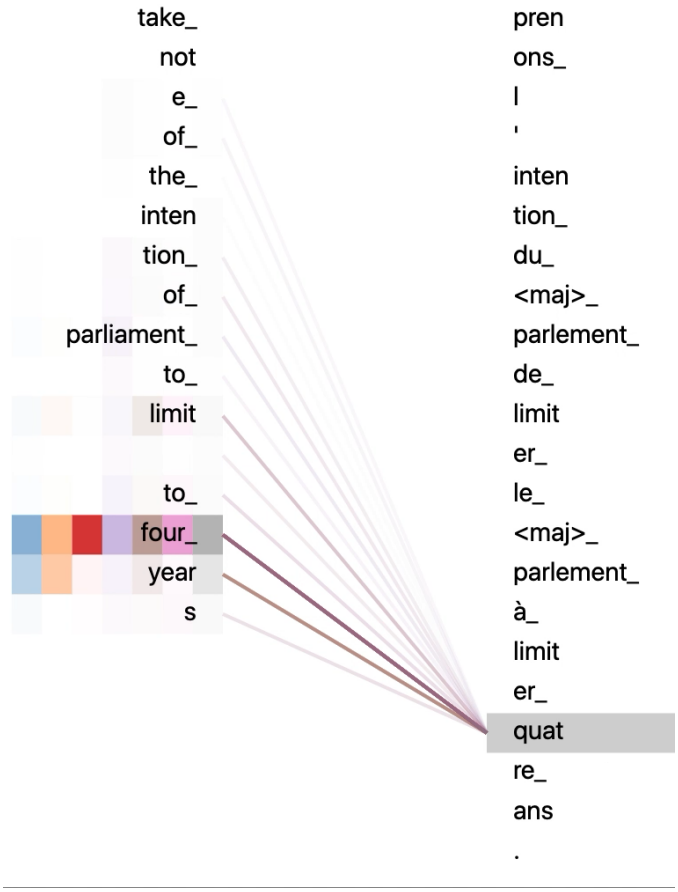


Figure 10: Animated[5] visualization of a sample EN→FR translation's attention scores from the last attention heads of our proposed Transformer model.

[5] The animation is only supported on some PDF viewers, notably Adobe's reader. If your reader does not support it, it will simply display the default frame for one of the subwords.