Low-Resource Neural Machine Translation

Fernanda Carmo, Raghav Gupta, Bhavya Patwa, Sabi Traifi
Université de Montréal, Québec, Canada

1 Introduction

In the last few years, we have witnessed a revolution in the field of Natural Language Processing (NLP). Seminal work in the field of deep learning and neural language modelling has led to drastic improvements in several NLP tasks, including Machine Translation. Today, Neural Machine Translation (NMT) systems power tools like *Google Translate* and provide state-of-the-art results, surpassing complex and hard-to-maintain traditional Statistical Machine Translation tools.

One key ingredient to the success of NMT has been the availability of massive amounts of data, both parallel sentences between pairs of languages (referred to as *bitext*) and monolingual data, required to train the models. While a huge quantity of data is widely available for mainstream languages (e.g. United Nations reports in many languages, Wikipedia pages etc), there are thousands of language pairs for which only a small amount of bitext can be found. In that context, NMT performs rather poorly, thereby leaving behind these less documented languages. *Low-Resource NMT* is a recent and active research area that aims at tackling this challenge.

In this study, we "simulate" a Low-Resource NMT scenario between English (the *source* language) and French (the *target* language), to analyze the latest deep learning techniques in a context with limited data, without relying on any pre-trained artifact (like models or embeddings). Our limited input data consists of:

- a bitext of 11K parallel sentences (each English sentence comes with a French translation). The given source sentences have been stripped from punctuation and lower-cased, while the target sentences are properly punctuated and capitalized.
- two distinct (unrelated) *monolingual* corpora, one in English and one in French, each having 474K unaligned sentences. Both are properly punctuated and capitalized.

We start with a supervised baseline (i.e. using only the limited bitext) and compare two leading NMT architectures: sequence-to-sequence (seq2seq) GRU model with Attention and Transformer model. We then explore the performance of two existing data augmentation techniques that make use of monolingual corpora: **back-translation** (using the target monolingual corpus) and **self-training** (using the source monolingual corpus). Finally, we propose a hybrid model that combines these two techniques and gets trained iteratively. We compare the performance of the different models and techniques using the SacreBLEU implementation [1] of BLEU (BiLingual Evaluation Understudy), the common evaluation metric for machine translation, based on n-gram precision.

Our study shows the benefits of back-translation and self-training and highlights the importance of hyper-parameter search and fine tuning in low-resource conditions.

The next sections are organized as follows: in section 2, we first provide an analysis of the input data, followed by a literature review in section 3. We then explain our methodology in details in section 4 and discuss our results in section 5. Finally, we conclude with a summary of our findings and areas for further work in section 6.

2 Data Analysis

Table 1 provides statistics on the vocabulary in the input data (all words have been lower-cased to compute these statistics). The coverage ratio provides the ratio of unique words in the parallel corpus that are found in the same-language monolingual corpus. We can see that the monolingual corpora cover more than 96% of the related parallel corpora. We also notice that, as expected from empirical experience, the French language uses a larger vocabulary. We highlight the lack of punctuation and capitalization in the source parallel corpus, which makes the translation task more complex.

In section 4, we will compare vocabulary representations based on words and sub-words.

Language	Corpus	Sentence count	Word count	Unique word count	Coverage ratio
en	parallel	11,000	216,360	13,658	-
fr	parallel	11,000	275,674	17,329	-
en	monolingual	474,000	9,294,343	60,009	96.75%
fr	monolingual	474,000	11,223,872	83,328	96.14%

Table 1: Statistics on input data

In terms of sequence length, as shown in Figures 1 and 2, we can see very similar distributions between parallel and monolingual corpora. Sentences in French tend to contain more words. These considerations are important to ensure that there is no domain mismatch between the bitext and the monolingual corpora. Going through these corpora shows that they come from the same sources. We can roughly categorize our data into 2 types of sentences from different domains, in both the bitext and monolingual corpora :

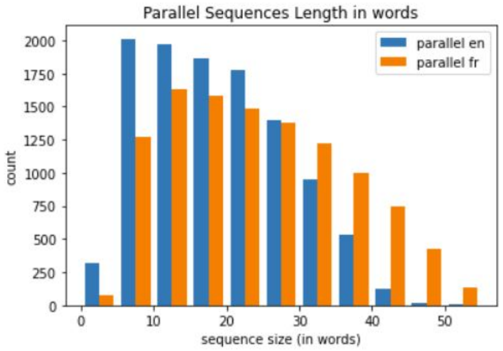

Figure 1: Parallel sentences seq. length

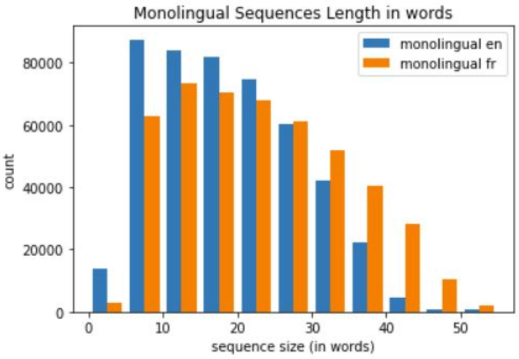

Figure 2: Monolingual sentences seq. length

- Parliament proceedings: using a formal style, most long sentences are in this category.
  example: *The Vienna European Council invited the Commission to authorise the Member States to apply the reduced rate on labour-intensive services having no cross-border content.*
- Conversational sentences (or speech transcription): usually shorter and using both formal and informal (or even slang) vocabulary:
  examples: *1. Can you watch the kids? 2. I have a week to do my homework. 3. This scares the shit out of me.*

While this mix of different topics and styles certainly adds complexity to the translation task, we can disregard the issue of domain mismatch, based on our observation that the bitext and monolingual corpora seem to come from the same sources of data.

3 Literature Review

3.1 Neural Machine Translation (NMT)

In the last few years, NMT has made huge progress to become the dominant technique for machine translation. NMT uses an encoder-decoder architecture, where an RNN-based encoder first produces an encoding of the source sequence. This learned representation is then fed to an RNN-based decoder that acts as a conditional language model to generate the translated sequence, in the target language. The model is trained using cross-entropy loss to maximize the probability of the next token in the output sequence. The performance of an NMT model is usually evaluated using the BLEU metric, which measures n-gram overlap between the generated sequence and the "ground truth" translation. The encoder and decoder modules are usually based on a GRU [3] or LSTM model [4] to better handle long-term dependencies.
One major breakthrough has been the introduction of the *attention* mechanism in a *sequence-to-sequence* (seq2seq) model, back in 2014 [2], which allows the decoder to selectively "shift its attention" to different parts of the source sequence during translation.

In 2017, an encoder-decoder model with multi-head self-attention and without recurrent layers, known as the *Transformer* model [5], reached state-of-the-art NMT performance while significantly reducing training time.
These models exhibit impressive results when trained with substantial amounts of bitext, but their performance drops significantly in low-resource conditions [6].

3.2 Semi-supervised techniques for Low-Resource NMT

To complement the limited bitext available in a low-resource setup, many techniques leverage monolingual data to improve the quality of the translation models [7]. Early approaches [8] consisted in using monolingual data to separately train a Language Model in the target language (doing next word prediction) and a Translation Model with the limited bitext, and then merge both at the decoder level.

Semi-supervised NMT techniques such as *back-translation* [9], using target-language monolingual data, and *self-training* [10], using source-language monolingual data, are data augmentation techniques that generate *synthetic bitext* that can be used to iteratively improve translation results [11]. Facebook AI Research team combined these techniques and used ensemble methods to get very promising results [12, 13]. Injecting some noise into the synthetic source sentences (word drop and/or swap) during training also improves generalization[10] [13]. Figures 3 and 4 below depict back-translation and self-training.

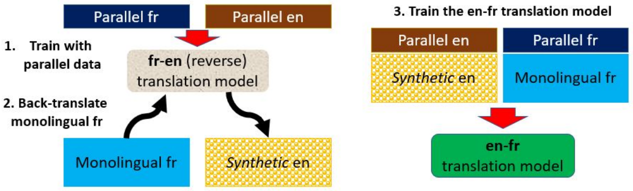
Parallel fr
Parallel en
1. Train with parallel data
fr-en (reverse) translation model
2. Back-translate monolingual fr
Monolingual fr
Synthetic en
3. Train the en-fr translation model
Parallel en
Parallel fr
Synthetic en
Monolingual fr
en-fr translation model

Figure 3: Back-translation

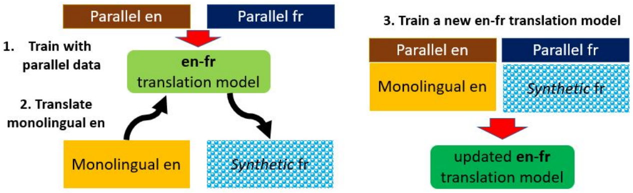
Parallel en
Parallel fr
1. Train with parallel data
en-fr translation model
2. Translate monolingual en
Monolingual en
Synthetic fr
3. Train a new en-fr translation model
Parallel en
Parallel fr
Monolingual en
Synthetic fr
updated en-fr translation model

Figure 4: Self-training

3.3 Transfer Learning

Multi-task learning with seq2seq models, combining a main translation task with auxiliary tasks such as constituency parsing [14] (or sentence auto-encoding [15]) and leveraging a single encoder, proved its usefulness in terms of transfer learning, even prior to the advent of attention [16]. It was also proven that adding an objective of *input sentences re-ordering* in low-resource conditions allowed better translations [10]. These techniques require careful fine-tuning of the mixing ratio between tasks during training. The addition of noise in input sequences also proves to be beneficial in unsupervised machine translation [17].

The *BERT* architecture [18] (from Google) is a Transformer-based encoder model that solves a Masked Language Model (MLM) task (i.e. predicting a masked word given its context) and can be pre-trained with unlabeled data to produce more robust representations. Fine-tuned on a specific task, BERT has demonstrated strong performance in natural language understanding tasks.

Inspired by BERT, the BART architecture [19] (from Facebook), based on the Transformer model, pre-trains a denoising sequence-to-sequence auto-encoder that corrupts sequences with noise and learns to reconstruct the original sequences. For NMT, the full BART model (encoder and decoder) is then used as a pre-trained target language decoder and fine-tuned using the limited bitext (the embedding layer of BART's encoder is actually replaced with a new, randomly-initialized encoder).

4 Methodology

In this section, we describe our experimental pipeline. Our approach can be divided into two major, sequential steps: (a) a **supervised baseline** using the bitext, where we assess two different architectures: a sequence-to-sequence GRU-based model with attention, and a Transformer model; and (b) **semi-supervised techniques** where we leverage monolingual data, by implementing self-training, back-translation and an iterative combination of the two. We also add noise to the input sequences in the form of word drop and word swap.

4.1 Supervised baseline

4.1.1 Models

Our study starts with the use and comparison of two major architectures for supervised NMT: a seq2seq model with Bahdanau attention [2] and GRU (inspired from [20]) and a Transformer model (inspired from [21]). The seq2seq model GRU uses recurrent layers and as such requires sequential training, which is time-consuming. On the other hand, the Transformer model does not use recurrence as it relies on self-attention mechanisms. This allows training to be parallelized, and therefore faster.

Considering the limited bitext we have, we hypothesize that flatter models will perform better than deeper ones, so we try different model sizes to verify this hypothesis. We provide details of the model configurations in section 5.

4.1.2 Experiments

We perform an 80-20% training-validation randomly shuffled split of the bitext. The validation set is fixed for all experiments for comparison and the scores in the results section 5 are reported on this validation set. The monolingual english and french texts are tokenized. Tokenized monolingual english text is further processed to remove punctuation and lower-case.

To handle different sequence lengths during training, we create mini-batches by padding zeros till the max-sequence length of the language. During calculation of loss, we use a mask to ignore the padded part of the sequence. We also assess the influence of the max-sequence length on the performance - truncated sequences vs max-sequence of the input language.

In terms of punctuation and capitalization, we leave it to the model to learn these patterns instead of using any post-processing technique or a separate learning model.

We use *Adam* optimizer [22], with the recommended learning rate setup in the seq2seq GRU model and the Transformer's custom learning rate schedule as described in [5]. The transformer adds positional encoding to the embedding to bring out a temporal relationship in the sequence. We use cross-entropy loss as our loss function.

We run 50 epochs with early stopping, and compare stopping criteria such as validation loss and validation accuracy. Ideally we would prefer early-stopping based on the BLEU score, but doing "generation", i.e. translating the validation set (inference) after each epoch, is very time-consuming. We decide to focus our efforts on better modelling techniques using greedy decoding, and leave engineering aspects like beam search for future work.

To compute the BLEU score, we generate the translations for the validation set and we use sacreBLEU evaluation script with the generated translations and the target sentences of the validation set.

For hyper-parameter tuning of the model, we vary the model size (number of layers, hidden dimensions), vocabulary size, token representation (words vs BPE), batch size, dropout rate, noising rate of input sequences and the maximum input sequence length.

We discuss the effects of these design choices and hyper-parameters on the performance of the model in subsection 5.1.

4.1.3 Vocabulary considerations

Once the best architecture from supervised baseline is found, we experiment with truncating vocabulary - full bitext vocabulary vs truncated vocabulary, different representations of vocabulary - word level vs sub-word level (BPE), and different word embeddings - randomly initialized vs static fastText [24] embeddings vs trainable fastText embeddings.

BPE [23] uses sub-word units representation instead of words which reduces the vocabulary size and the number of *unknown* tokens, but lengthens the sequence length of the sentence. In our specific case of limited data and language closeness (same latin alphabet), we want to find out which representation performs best - word or BPE.

While creating fastText embeddings, we make use of the large monolingual corpora. Static fastText embeddings, which are used instead of a model's embedding layer, have the advantage that the model will never see an unknown token, otherwise the vocabulary has to be fixed to create the lookup table for embeddings layer of a model. Trainable fastText embeddings are implemented by initializing the lookup table of the model's embedding layer with the fastText embeddings, rather than random initialization.

4.2 Semi-supervised techniques

We now focus on leveraging the monolingual data in order to boost the performance of our best supervised model using data augmentation techniques. Since the size of synthetic bitext dataset is much larger than the parallel bitext, we train the model for each epoch by either up-sampling the parallel bitext data by a repeat factor of $\frac{\text{synthetic bitext batches}}{\text{parallel bitext batches}}$, or by down-sampling the synthetic bitext to the same number of batches as parallel bitext, randomly shuffling the synthetic bitext after each epoch.

For **self-training**, we use our supervised en-fr model to translate a subset of the English monolingual data. We then train a new transformer model, first with this *synthetic* bitext and then with the real bitext (as done in [12]) in an epoch. We perform experiments with different ratios of synthetic bitext vs real bitext. We also experiment with **noise injection**, only in the synthetic bitext source sequences, as follows: every source sequence has a probability P_{seq}^{drop} of having dropping applied and a probability P_{seq}^{swap} of having swapping applied. In case of dropping, each token of the sequence has a probability P_{tok}^{drop} of being dropped. In case of swapping, one randomly selected token in the sequence is swapped with a contiguous token. Due to time constraints, we limit our noise experiments to these two options, although more sophisticated noise techniques could be tried (see for example [17]).

As described in subsection 3.2 of the literature review, we adapt the **back-translation** method [9]. To do so, we first need to build a *reverse* fr-en translation model using the bitext. We train a fr-en model, using the same 80-20% split. (In section 5, we also report the BLEU score of this supervised fr-en translation model). With this reverse model, we then back-translate our monolingual French data into *synthetic* English sentences. We finally use this *synthetic bitext* to retrain our best en-fr supervised model. We use less aggressive noise injection in synthetic input sequences than in self-training.

Finally, we run an experiment by training both synthetic bitexts obtained by self-training and back-translation, to see whether the combination further improves the BLEU scores. We then iterate this process by using the best model found to create new better *synthetic bitext*, an approach adapted from Facebook AI research [13].

5 Results and Discussion

5.1 Supervised Baseline

We compare here the performance of the two architectures (GRU-based seq2seq with attention and Transformer) in a supervised context, using the bitext (80% for training and 20% for validation). The GRU-based seq2seq model uses embeddings of size 256, an output dimension of size 512, a Bahdanau additive attention layer of 8 units, a batch size of 64 and no dropout. We vary the number of GRU layers as per Table 2. We quickly realized that the Transformer model gives superior results, so we focused our hyper-parameter search on the Transformer model. We tried limiting the word vocabulary size in both languages and architectures, but did not see any improvement, so we used the full vocabulary. This was expected considering the limited bitext. Tables 2 and 3 below show the main results we observed.

S.No.	model	# of layers	max seq len	BLEU
1	seq2seq GRU with attention	2	-	2.37
2		1	-	2.51
3		1	30	2.44
4		1	40	2.57

Table 2: Supervised GRU seq2seq with attention BLEU scores

The GRU model performs rather poorly, unable to generalize with such a low amount of data.

The base transformer (S.No. 1) is based on the recommended configuration of hyper-parameters of the transformer model [5]. We notice that a lesser number of layers and more aggressive dropout performs much better (S.No. 2-11). Reducing the batch size to 8 decreased the performance significantly (S.No. 3), hence we kept it at 64. Reducing the transformer to 1 layer each of encoder-decoder performed the best, but increasing the width, i.e. the hidden dimension size of the layer (d_{model} = hidden dimension of the embedding and of the attention output; d_{ff} = hidden dimension size of fully connected layer) led to better results (S.No. 4-7,9-11) till 1024, then performance starts degrading on increasing further to 2048 (S.No. 8). We note that limiting input sequence lengths to a maximum of 40 word tokens gives the best BLEU score. We see here the importance and sensitivity of hyper-parameter fine tuning in a low-resource setup.

S.No.	model	# of layers	d_{model}	d_{ff}	dropout	Batch Size	max seq len	BLEU
1	transformer	6	512	2048	0.1	64	-	4.44
2		2	256	256	0.4	64	-	7.67
3		2	256	256	0.4	8	-	4.01
4		1	128	128	0.4	64	-	7.95
5		1	256	256	0.4	64	-	9.01
6		1	512	512	0.4	64	-	8.98
7		1	1024	1024	0.4	64	-	**9.73**
8		1	2048	2048	0.4	64	-	8.51
9		1	1024	1024	0.5	64	30	10.37
10		1	1024	1024	0.5	64	35	10.45
11		1	1024	1024	0.5	64	40	**10.73**

Table 3: Supervised Transformer model: hyper-parameter search. Model S.No. 7 was used for future experiments because transformers with max sequence length were experimented much later.

We noticed a correlation between the validation loss and BLEU score, hence we were early stopping based on best validation loss.

Please note that Transformer S.No. 7 in Table 3 was used for further experiments as we continued fine-tuning the hyper-parameters of the supervised model till the end (the best supervised model being Transformer S.No. 11).

We select one example of translation to compare the 2 models in Table 4. The seq2seq GRU model aborts the translation very quickly, while the transformer model better attempts to capture the meaning of words that appear late in the input sequence.

source	it is not clear what the writer is trying to say
seq2seq GRU (S.No. 2)	Il ne peut pas .
transformer (S.No. 7)	Il ne s' agit pas clairement clairement clairement à dire que c' est dire à dire .

Table 4: Translation comparison between seq2seq GRU and transformer.

5.2 Vocabulary and Embedding experiments

The previous experiments were done using a vocabulary based on words. Using our best model, we also ran several experiments using BPE sub-words and fastText embeddings, but none of them gave better results than using a word vocabulary. Table 5 lists the techniques and results we observed. Static fastText embeddings perform poorly as they are not trained with the transformer and it might make it harder for the transformer to fit the weights based on them. Hence, using static fastText embeddings, for the advantage of no unknown tokens, is not worth it. Even trainable fastText leads to worse results than randomly initialized embeddings. This may be because the semantic meaning captured from fastText embeddings are not suitable for the transformer hidden representation and would instead degrade the training, leading to bad convergence. We also notice that thresholding vocabulary size leads to an huge amount of unknown tokens in the dataset and the model overfits, predicting unknown tokens most of the time, leading to a bad BLEU score.

BPE after thresholding codes performed similar but slightly worse than word level, probably because the sentence length becomes larger with subwords and becomes harder for the transformer to generate the right subwords in sequence. Transformer performance is negatively correlated with sequence length, as described in subsection 5.5.

technique	BLEU
Vocab trimming (Eng 10k, French 12k)	2.73
Static fastText embeddings	1.94
Trainable fastText embeddings	3.23
BPE (no thresholding on BPE pairs)	6.89
BPE pairs thresholded at 10,000	9.06
Word level, no vocab trimming	**9.73**

Table 5: Vocabulary experiments

5.3 Semi-supervised approaches

We now leverage monolingual data using self-training and back-translation, with different ratios of synthetic data to real bitext, and a combination of both. Table 6 lists the techniques and BLEU score we obtained.

For both self-training and back-translation, we added noise to the input sequences: every input sentence, excluding the real bitext, is noised with either word swapping (50%) or word dropping (50%). In case of word swapping, 2 contiguous words are swapped at a random position in the sequence. In case of word dropping, each word in the sequence has a probability of being dropped, of 0.3 for self-training and 0.1 in back-translation. Ideally hyper-parameter search should be applied to find the best ratios, but due to limited amount of time and resources, we used common values found in the literature [12] [13].

For back-translation, we first trained a fr-en supervised model that got a BLEU score of *10.67*. This score is higher than in the en-fr direction and understandable because the English text is not punctuated nor capitalized, so higher n-gram precision is easier to achieve.

technique	**iteration**	**# of mono sentences**	**ratio**	**BLEU**
Self-training (ST)	-	8,800	1	8.54
	-	17,600	2	9.31
	-	35,200	4	8.73
Noisy self-training (NST)	-	8,800	1	8.35
	-	17,600	2	9.15
	-	35,200	4	9.51
	-	176,000	20	11.16
Back-translation (BT)	-	8,800	1	9.41
	-	17,600	2	9.76
	-	35,200	4	10.01
	-	176,000	20	11.8
Noisy back-translation (NBT)	-	8,800	1	8.06
	-	17,600	2	9.09
	-	35,200	4	9.70
NST + BT (down-sampling)	1	176,000	20	12.57
NST + BT (up-sampling)	1	176,000	20	12.86
NST + BT (up-sampling)	2	176,000	20	**15.51**

Table 6: Performance with semi-supervised techniques

We can see that back-translation performs better than self-training and this is expected because back-translation introduces synthetic data with real target language sentences, while self-training relies on synthetic target language sentences. We reach better scores when a higher ratio of synthetic data is being used for training.
In terms of noise, we see that noising input sequences in self-training is beneficial, while back-translation performs better without noise. This makes sense because in self-training, synthetic input sentences are proper english sentences (so noise can help generalize better), while in back-translation, the synthetic input sequences have been back-translated from french monolingual data, and as such they are already very noisy.

We then use a ratio of 20 between synthetic bitext and real bitext, for noisy self-training and back-translation and see BLEU score improvements. We also observe that a combination of both self-training and back-translation techniques is beneficial.

We noticed that validation loss was no longer in correlation with BLEU score for these models trained on larger data, and instead saw a strong correlation with validation accuracy and BLEU score, and hence were early stopping based on validation accuracy. The additional data in the semi-supervised setting allowed to train a bigger model as compared to the supervised setting.

For the second iteration, we use the best model from iteration 1 for self-training, and for back-translation, we trained another fr-en model using the parallel and *synthetic bitexts* (BLEU score 13.37), to generate better *synthetic bitexts*. Our best performing model is second iteration Noisy ST + BT with up-sampling parallel bitext, Transformer with 2 layers, 1024 size embedding, 1024 hidden dimension after attention and 1024 nodes in fully connected, giving a BLEU score of 15.51. The generated *synthetic bitexts* improves in each iteration as they generate with better models. Ideally the process could be repeated till convergence (BLEU score stops increasing), but this is a time and resource consuming procedure (so we had to stop at the second iteration).

We also analyze the capability of the model to capitalize, getting an accuracy of 66%, and punctuate, being accurate 47% of the time, which is fully learned by the model. The lack of capitalization can be because most proper nouns in the validation set might not be in vocabulary, hence it does not give the correct output translation. Better results could probably have been achieved using multi-task learning by training a separate task that learns punctuation and capitalization, using the monolingual data, and by training on a character-level or sub-word level.

5.4 Examples of translations

We provide in Table 7 some hand-picked examples of interesting (and sometimes hilarious) translations of our best model. The model is clearly influenced by the formality of parliament proceedings from the training data and this bias directly impacts translation: we sometimes see politically-correct "translations" of colloquial text. The model seems to have memorized chunks of contiguous tokens during training, sometimes leading to very creative translations (e.g. in bold). We also see that longer sentences generate more confused translations, which often repeat words or series of words. We tried a post-processing step that alleviates this tendency by removing contiguous repetitions of tokens, and it gives a 1% increase in BLEU score (from 15.51 to 15.67). The overall translation quality remains however low, highlighting the difficulty of low-resource NMT.

<table>
<tr><td>he ’s very sexy
Il est très intelligent .</td><td>nobody talked about europe
Personne n’ a parlé de l’ Europe .</td><td>that is why i abstained
C’ est pourquoi je me félicite de mon vote .</td></tr>
<tr><td>i think so too
Je pense donc trop .</td><td>money rules the world
Les règles du monde entier .</td><td>which ice cream shop are you going to
Quels sports aimes - tu ?</td></tr>
<tr><td colspan="2">computers are capable of doing extremely complicated work
Des objets inanimés qui sont extrêmement sérieux .</td><td>and everyone ’s somewhere in the middle
Et tout le monde est quelque part au Moyen - Orient .</td></tr>
<tr><td colspan="2">mr president ladies and gentlemen first of all i should like to congratulate mr lyon on his work and also the chair and the secretariat of the committee on agriculture and rural development
Monsieur le Président , Mesdames et Messieurs , je voudrais tout d’ abord féliciter M. Garriga Polledo pour son travail et le comité de développement de l’ agriculture et du comité de développement rural , ainsi que le comité de l’ agriculture .</td><td>finally in 1999 in tampere the member states came out with this grand statement that they wanted a common asylum and immigration policy
Enfin , en 1999 , dans l’ espace des États membres ont pris cette déclaration pour faire une déclaration commune qu’ ils avaient une politique commune commune commune commune commune et l’ immigration commune .</td></tr>
</table>

Table 7: Hand-picked translation examples from the best model.

5.5 Impact of sequence length on BLEU score

We now use our best model to analyze the impact of sequence lengths on the BLEU score. Figure 5 shows the average BLEU score based on the length of the target validation sequences. As expected, the BLEU score degrades when the sequence length increases. While the average BLEU score is 15.51, it increases to 19 for short sentences up to 10 words and it is around 12 for sequences longer than 40 words.

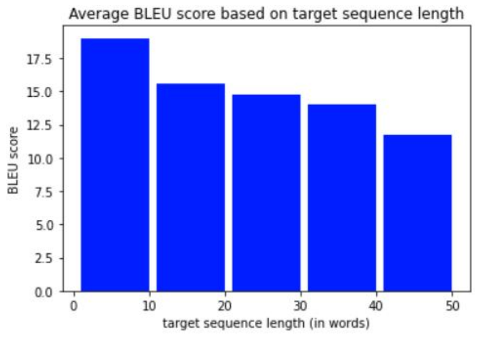

Figure 5: BLEU score vs target sequence length.

6 Conclusion

In this study, we have proposed an approach to handle a low-resource NMT task. We first used supervised learning, based on the bitext, and compare two major architectures: a GRU-based seq2seq model with attention and a Transformer model, using a word-based vocabulary.

After establishing the superiority of the Transformer model, we tried a sub-word vocabulary based on BPE, and fastText for trained embeddings, and got limited results. We noticed that limiting the input sequence length improves BLEU scores and highlighted the importance and sensitivity of hyper-parameter fine tuning in a low-resource context. We observed that the model learns to handle punctuation and capitalization on its own, to a certain extent. We then leveraged the monolingual data using self-training (using source monolingual data) and back-translation (using target monolingual data). We showed that these complementary techniques can improve translation performance when applied in an iterative manner. We also showed that noising the input sequences in self-training, with random word drop and swap, is a good regularization technique. One important aspect that requires further study is the optimal ratio of synthetic bitext and real bitext, both in terms of quantity and mini-batch sampling during training. We note as limitation that that because our models work at word level, they cannot translate words that are out of vocabulary. We also expect beam search decoding could have helped boost performance.

As future work, we would like to experiment with transfer learning techniques that pre-train Transformer models using monolingual data and masking techniques, such as in BART [19]. We could also experiment a multi-task setup that jointly learns the translation task and a punctuation/capitalization task that leverages monolingual data.

References

[1] Post. 2018. *A Call for Clarity in Reporting BLEU Scores.* https://arxiv.org/abs/1804.08771

[2] Bahdanau et al. 2014. *Neural Machine Translation by Jointly Learning to Align and Translate.* https://arxiv.org/abs/1409.0473

[3] Cho et al. 2014. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* https://arxiv.org/abs/1406.1078

[4] Hochreiter & Schmidhuber. 1997. *Long Short-Term Memory.* https://doi.org/10.1162/neco.1997.9.8.1735

[5] Vaswani et al. 2017. *Attention Is All You Need.* https://arxiv.org/abs/1706.03762

[6] Koehn & Knowles. 2017. *Six Challenges for Neural Machine Translation.* https://arxiv.org/abs/1706.03872

[7] Gibadullin et al. 2019. *A Survey of Methods to Leverage Monolingual Data in Low-resource Neural Machine Translation.* https://arxiv.org/abs/1910.00373

[8] Gulcehre et al. 2015. *On Using Monolingual Corpora in Neural Machine Translation.* https://arxiv.org/abs/1503.03535

[9] Sennrich et al. 2015. *Improving Neural Machine Translation Models with Monolingual Data.* https://arxiv.org/abs/1511.06709

[10] Zhang & Zong. 2016. *Exploiting Source-side Monolingual Data in Neural Machine Translation.* https://www.aclweb.org/anthology/D16-1160

[11] Hoang et al. 2019. *Iterative Back-translation for Neural Machine Translation.* https://www.aclweb.org/anthology/W18-2703

[12] He et al. 2020. *Revisiting Self-training for Neural Sequence Generation.* https://arxiv.org/abs/1909.13788

[13] Chen et al. 2019. *Facebook AI's WAT19 Myanmar-English Translation Task Submission.* https://arxiv.org/pdf/1910.06848

[14] Currey & Heafield. 2019. *Incorporating Source Syntax into Transformer-Based Neural Machine Translation.* https://www.aclweb.org/anthology/W19-5203

[15] Cheng et al. 2016. *Semi-Supervised Learning for Neural Machine Translation.* https://arxiv.org/abs/1606.04596

[16] Luong et al. 2015. *Multi-task Sequence to Sequence Learning.* https://arxiv.org/abs/1511.06114

[17] Lample et al. 2018. *Unsupervised Machine Translation Using Monolingual Corpora Only.* https://arxiv.org/abs/1711.00043

[18] Devlin et al. 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* https://arxiv.org/abs/1810.04805

[19] Lewis et al. 2019. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.* https://arxiv.org/abs/1910.13461

[20] TensorFlow tutorial: *Neural machine translation with attention.* https://www.tensorflow.org/tutorials/text/nmt_with_attention

[21] TensorFlow tutorial: *Transformer model for language understanding.* https://www.tensorflow.org/tutorials/text/transformer

[22] Kingma & Ba. 2014. *Adam: A Method for Stochastic Optimization.* https://arxiv.org/abs/1412.6980

[23] Sennrich et al. 2016. *Neural Machine Translation of Rare Words with Subword Units.* https://arxiv.org/abs/1508.07909

[24] Bojanowski et al. 2016. *Enriching Word Vectors with Subword Information.* https://arxiv.org/abs/1607.04606