Low Resource Machine Translation

IFT6759 - Advanced Projects in Machine Learning

Olivier Tessier-Lariviere
Mila, University of Montreal

Azfar Khoja
Mila, University of Montreal

Ryan Mokarian
Mila, University of Montreal

Yichen Lin
Mila, University of Montreal

Abstract

Adequate parallel data for training a Neural Machine Translation system is not always available. In this work, we explore strategies to increase machine translation performance in a low resource scenario through the use of generally more widely available monolingual data. We leverage target monolingual data using back-translation and improve our baseline score by 3.5 BLEU. We find that the use of pre-trained embeddings on the source monolingual data is ineffective.

April 30, 2020

1 Introduction

Machine translation systems typically require a large amount of parallel data to obtain great performance. Parallel data is usually organized into pairs of sentences in the source and target language. However, a large number of parallel examples are not always available in every language pairs.

In this work, we explore ideas that help models generalize despite a small amount of parallel training data. To achieve this, we leverage monolingual data. Monolingual or unaligned data means that sentences in one language do not have their translated version in the other language. As one might expect, the amount of available monolingual typically far exceeds the amount of parallel data.

More specifically, the goal of this project is to implement a machine translation system that can translate from English to French in a low resource scenario. The quality of the translation will be evaluated according to the BLEU metric. This metric computes a similarity score between a translated sentence and a reference sentence. The score lies between 0 and 100 (or 0 and 1). The more similar the text, the higher the score will be. It is important to note that even human translation does not attain a perfect score since that would indicate that the translation is identical to the reference. The BLEU score can be computed using multiple references, but in this setting we only have access to one. To get an estimate of the quality of the translation on the corpus, the sentence scores are averaged.

The report begins with a "Data analysis" section, which describes the data and provides some insights. This is followed by a literature review that covers the various solutions to deal with low resources in machine translation. The subsequent section is "Methodology", which describes our pre-processing, explains our selected models and their architectures as well as the selected strategies to deal with low resources. The final sections present our experimental results, analysis and conclusions.

2 Data analysis

The data consists of an aligned corpora of 11k sentences in English and French as well as an additional monolingual corpora of 474k sentences in each language. In the aligned data, the source language (English) lacks formatting. It has been stripped of punctuation marks such as commas, periods and capital letters. On the other hand, the target language (French) is properly formatted. Both monolingual corpora are also properly formatted.

The content of the data seems to be a transcript of parliament proceedings from the European Union. Sentence samples from the aligned data are shown in Figure 1. Although the text is mainly about politics, it is surprisingly varied. Indeed, the conversations can range from nuclear weapons to a banana trade dispute.

Source: *we are particularly dependent on the continuation of an eu system of export subsidies in order to remain competitive on export markets*
Target: *Nous sommes particulièrement dépendants du maintien du système d' aides à l' exportation de l' ue , en vue de rester compétitifs sur les marchés à l' exportation .*

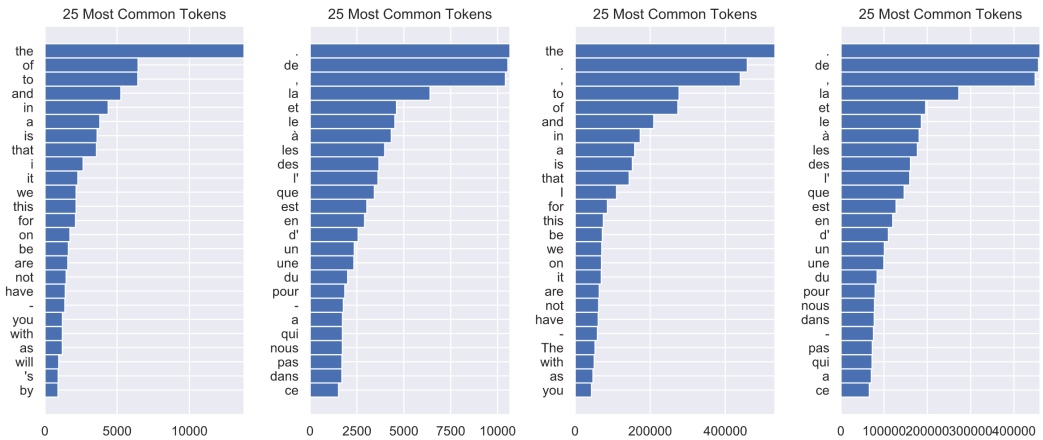
Source: *the european union must be coherent and respect the legislation that we pass through this house*
Target: *L' Union européenne doit faire preuve de cohérence et respecter la législation que nous adoptons dans cette enceinte .*

Source: *what do you learn spanish for*
Target: *Pourquoi est - ce que tu apprends l' espagnol ?*

Figure 1: Sentence samples from the aligned corpora.

The aligned English data contains 13657 unique words whereas the unaligned English data contains 69903 unique words. This disparity continues in the french language, where the aligned data contains 18235 unique words and the unaligned data contains 90509 unique words. This shows that the unaligned corpora has a vocabulary about 5 times larger than the aligned one.

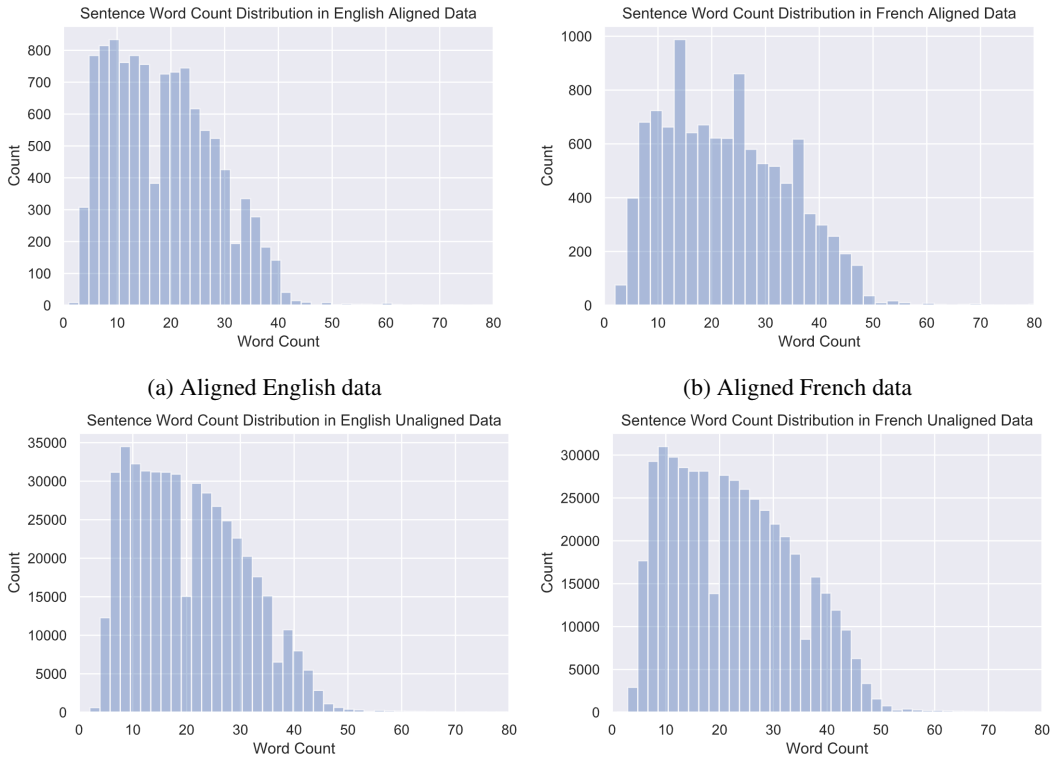
Figure 2 shows the most common tokens in the different corpora. As we can see, the aligned and unaligned french are very similar. The most common 18 tokens are exactly the same. The aligned and unaligned English are also very similar, except for the fact that the aligned English data does not contain punctuation.



(a) Aligned English (b) Aligned French (c) Unaligned English (d) Unaligned French

Figure 2: 25 most common tokens in the different corpora.

Figure 3 shows the word count per sentence in the aligned data. Here, the punctuation tokens such as periods and comma counts as a word. As we can see, the two languages follow a similar distribution with most of the sentences falling between 2 and 50 words. Aligned English sentences are a little bit shorter but this may be due to missing punctuation.



(a) Aligned English data

(b) Aligned French data

(c) Unaligned English data

(d) Unaligned French data

Figure 3: Distribution of the number of words per sentence in the aligned data.

3 Literature review

In the following paragraphs, different approaches of Machine Translation (MT) are presented and followed by Neural Machine Translation (NMT). Then, we provide an overview of the existing high-performance NMT architectures such as word embeddings, sequence-to-sequence models, attention mechanisms and transformer models. Subsequently, pre-training, back-translation and multi-task learning are described as transfer learning strategies to deal with the low-resource scenario. Finally, a metric to evaluate the quality of machine-translated text is presented.

MT is a sub-field of computational linguistics. It uses a computing device to automatically perform translation of a natural language from one language to another. MT approaches are categorized into Rule-Based Machine Translation (RBMT), Statistical Machine Translation (SMT), Example-Based Machine Translation (EBMT), Hybrid MT and NMT [13]. Earlier research focused on RBMT. The RBMT paradigm relies on built-in linguistic rules to transfer the grammatical structure of the source language to the target language using a parser and an analyzer. SMT, on the other hand, utilizes statistical translation models to implement predictive algorithms to teach a machine how to translate text. EBMT is the idea of translation by analogy. The corpus that is used is one that contains texts that have already been translated and phrasal translations are translated by analogy to previous translations. Hybrid MT is a method that uses the strengths of rule-based and statistical translation practices [3].

NMT emerged and became a major area of research with the popularity of deep neural networks in 2012 [3]. NMT needs only a fraction of the memory required by traditional SMT. Its results are promising and have recently made fast advancement. As an example, Google has switched from SMT to NMT for its translation services [13].

Word embedding is a technique that helps NMT achieve better performance. It is an approach to distribute the representation of words in a vector space across all available feature space dimensions. It groups similar words and find linguistic regularities and patterns. Word embedding improves the quality of the learned word and phrase representations, especially for the entities that are rare [6].

Sequence-to-sequence (seq2seq) approach is one of the most popular systems used in NMT. It is also known as the encoder-decoder architecture. This system was developed by Ilya Sutskever [11]. The main advantages of the encoder-decoder framework are the ability to train a single end-to-end model directly on source and target sentences and to qualitatively capture linguistic regularities at the word level as well as phrase level [2]. The downside of seq2seq model is that translation quality drops significantly when an NMT system translates long sentences. The reason is that seq2seq model requires fixed-sized input and output vectors, i.e. all the information of a source sentence needs to be compressed into a fixed-length vector representation which cannot identify all the cues for the decoder to generate appropriate translations. The concept of an attention mechanism is a method that helps with long sequences [1].

Attention mechanism, as opposed to the seq2seq model, does not attempt to encode the input sentence into a single fixed-length vector. It only focuses on the relevant parts of the source sentence while translating. More specifically, the input sentence is encoded into a sequence of vectors and a subset of these vectors is adaptively chosen while decoding the translation [1]. In the attention mechanism, this adaptation is performed through accessing the decoder to the hidden states of the encoder. Each hidden state has a weight indicating its level of importance with respect to the current output prediction of the decoder. Then a weighted sum of hidden states is passed to the decoder to use for prediction of the next word. When only a single attention weighted sum of the values is computed, capturing different aspects of the input would be more difficult. To solve this problem, the Transformer architecture emerged. This architecture, which introduced by Vaswani et al [12], is an extension of an attention mechanism when "multi-head attention" blocks are used.

Transformer architecture uses multi-head self-attention. Each self-attention mechanism is called a head and is meant to identify all elements in a sequence which are relevant to an element t. Multi-head self-attention can capture multiple relevant information, i.e. different properties about the input. For brevity, we refer readers to the Vaswani et al paper, [12]. The Transformer approach has demonstrated a high performance in NMT tasks, compared to the seq2seq with attention approach [12]. Currently transformer is the dominant NMT architecture choice and has been shown effective in large data scenarios [5]. This project deals with NMT in a low-resource scenario. While the transformer approach delivers a high performance in NMT tasks, it is often less effective for low-resource languages [4]. To solve the issue of data shortage in low-resource NMT, transfer learning approach is used. Pre-training, back-translation and multi-task learning are three transfer learning techniques that are explained in the following paragraphs.

Pre-training is one of the transfer learning techniques to solve the issue of data shortage in low-resource NMT. Zoph et al [18] are one of the first implemented this method. The main idea is to first train an NMT model on a dataset with high-resource language pair dataset (the parent model). Then, to initialize a new NMT model with the already-trained parent model and train it on a dataset with very little bilingual data, i.e. the low-resource pair (the child model). This means that the low-data NMT model will not start with random weights, but with the weights from the parent model.

Back-translation is a prominent transfer learning approach to deal with resource-poor scenarios and to alleviate the need for large parallel data. This technique generates synthetic bitext by translating monolingual sentences of the target language into the source language with a pre-existing target-to-source translation system. These noisy source translations are then incorporated to train a new source-to-target NMT system. The production of synthetic parallel texts is comparable with data augmentation methods used in computer vision [10].

Multi-task learning is another transfer learning technique. It simultaneously trains models on multiple learning tasks while sharing parameters across tasks [14]. By training a model not only on the machine translation task, but also on other Natural Language Processing tasks, the model can take advantage of inductive transfer between the tasks and to obtain an improved generalization over the translation performance [9] [7]. This performance improvement has been demonstrated for experiments with low-resource translation scenarios [16] [17].

In this project, we use BLEU (BiLingual Evaluation Understudy) as our evaluation metric. The BLEU score was proposed by Kishore Papineni, et al. in their 2002 paper "BLEU: a Method for Automatic Evaluation of Machine Translation" [8]. This evaluation metric is quick and inexpensive to calculate, easy to understand, language-independent and highly correlated with human evaluation. BLEU's output is a number between 0 and 1 indicating how similar a "candidate" machine translated text is to a "reference" human translated text, where 1 represents a perfect match [8].

4 Methodology

In this section, we present our experimental pipeline. The first step is to pre-process the different corpora. This is explained in section 4.1. Next, in section 4.2, we go over the two models we trained on the aligned data to obtain our baselines. Then, we present two different strategies to increase the performance of our best baseline using unaligned data. We explore pre-trained embeddings on the unaligned source data (English) in section 4.3 and we explore back-translation on the unaligned target data (French) in section 4.4. Finally in 4.5 we introduce beam search with length normalization.

4.1 Data Processing

The four corpora are processed before being fed to the machine learning algorithms. Word tokenization is performed on the text, meaning that each word corresponds to a token. Punctuation such as periods and comma also count as a token. We only consider words that are in the aligned corpora as part of the vocabulary.

A beginning of sentence token ($< start >$) and an end of sentence token ($< end >$) are added for each sentence. Words beginning with a capital letter are replaced with a special capital token ($< maj >$) followed by the lower-cased word. Similarly, words that are upper-case are replaced with a special upper-case token ($< upp >$) followed by the lower-cased word. The words that are not in the vocabulary are attributed an unknown token ($< unk >$).

The aligned data is split into a training set (85%) and a validation set (15%). The aligned English is the input and the aligned French is the target. The examples are sorted by sequence length to increase training efficiency by reducing the padding in each batch. The examples are padded with zeros.

4.2 Baselines

The first step of our experiments is to obtain a good baseline model using aligned data. The goal is to improve it later on using unaligned data. We do a hyperparameter search on two models.

The first model is an encoder-decoder GRU with attention. Attention is a mechanism combined in the RNN allowing it to focus on certain parts of the input sequence when predicting a certain part of the output sequence, enabling easier learning and of higher quality. A combination of attention mechanisms enabled improved performance in many tasks making it an integral part of modern RNN networks. We do a hyperparameter search to get the best BLEU on the validation set. We vary the embedding dimensions, the number of hidden units and the number of layers in the encoder and decoder.

The second model is a transformer. Transformers are designed to handle ordered sequences of data. However, unlike RNNs, transformers do not require that the sequence be processed in order. They do not need to process the beginning of a sentence before it processes the end. Due to this feature, the transformer allows for much more parallelization than RNNs during training. Again, we do a hyperparameter search to get the most performance out of this model. We vary the number of hidden units, the number of attention heads and the number of layers.

Both models are trained using the Adam optimizer and the cross-entropy loss. The padding is masked when computing the loss. The transformer uses the learning rate scheduler defined in the original transformer paper attention is all you need [12], while the encoder-decoder GRU uses a learning rate of 0.001. The transformer is trained for 100 epochs. Since the encoder-decoder GRU trains slower, it is trained for 30 epochs. The models are saved at the epoch that has the best validation BLEU score.

4.3 Pre-trained Embeddings

To exploit the abundant unaligned source (English) data, we explore learning dense word embeddings during a pre-training phase. Since the task of learning an embedding is self-supervised, we do not need labels. Models will typically learn an embedding in their first layers when training on the main task. Here, the idea is that better word embeddings can be learned on the unaligned source data because of the huge amount of available examples compare to the aligned source data.

We train FastText embeddings on the tokenized unaligned English data. FastText embeddings are continuous word representations. Contrary to most popular models that learn these representations by assigning a distinct vector to each word, FastText considers the word morphology. This is an advantage in our low-resource scenario because some words are rare and thus FastText can rely on similar words to learn a good representation.

Once the FastText model has been trained, we create a lookup embedding matrix. We only consider the embeddings of words that are in the aligned source corpus. Then, we initialize the embedding layer of our model with the embedding matrix. We first train our model with the embedding layer weights frozen. Then, we try to increase the model performance by fine-tuning the embedding layer. To do this, the model is trained with the embedding layer frozen for the first half of training and unfrozen for the second half.

4.4 Back-Translation

To exploit the abundant unaligned target data (French), we explore the use of back-translation to augment the number of examples in our training set. This method consists of first training a target to source (French to English) model on the aligned data. This model is then used to translate the unaligned target data, effectively creating synthetic source data. During training, we mix the newly created synthetic parallel data with the original parallel data. In the original back-translation work [10], they mix these two data sources in the ratio 1-to-1. In this work we explore different ratios. The ratio is computed using equation 1.

$$\text{back-translation ratio} = \frac{\text{number of synthetic examples}}{\text{number of original examples}} \tag{1}$$

We define an epoch as one iteration through the parallel data and we re-sample from the synthetic data at every epoch. We shuffle the two data sources together. Since the vocabulary consists of the words in the aligned data, many words coming from the unaligned data are attributed the unknown token. This causes the model to learn to generate the unknown token. To counter this, we generate the second more probable token when the model prediction is the unknown token.

4.5 Beam Search

The predictions at the decoder are generated in an auto-regressive manner, where we sample the most likely word at each step (greedy search). However, our goal is to generate the most likely sentence, which amounts to maximizing the joint likelihood of all steps. This can be expressed as

$$\text{argmax}_y \prod_{t=1}^{T_y} P(y_t|x, y_1, y_2, ..., y_{t-1})$$

where is x is our source sentence and y_t is the output word generated at time t.

Beam search is a heuristic search algorithm that keeps a set of k promising nodes at each step and explores the graph through each node in this set. k is a hyperparameter that balances computational complexity vs performance. We implement beam search using log-likelihood and length normalization as described in [15]. The final expression is represented below, here α is another hyper-parameter of the algorithm.

$$\text{argmax}_y \frac{1}{{T_y}^{\alpha}} \sum_{t=1}^{T_y} \log P(y_t|x, y_1, y_2, ..., y_{t-1})$$

5 Results

In this section, we present the results for the baselines (5.1), the pre-trained embeddings (5.2), the back-translation (5.3) and beam search (5.4). This is followed by a discussion in section 5.5 where we also show qualitative results.

5.1 Baselines

Table 4 shows the BLEU score of different models trained on the aligned data only. The results of the hyperparameter search for the encoder-decoder GRU and for the transformer are presented respectively at table 5 and table 6. Furthermore, the learning curves are available at figure 9 and 10.

Table 1: BLEU Score of Models Trained on Aligned Data Only

Model	BLEU	Loss
Encoder-Decoder GRU	3.86	**12.01**
Transformer	**8.42**	12.67

5.2 Pre-trained Embeddings

We take the best performing model from section 5.1 and try different pre-trained embeddings. Table 2 shows the BLEU score for the transformer for the different embeddings.

Table 2: BLEU Score using Pre-trained Embeddings

Model	Embedding	Dim	BLEU	Loss
Transformer	FastText	128	6.16	12.22
Transformer	FastText	256	6.53	12.32
Transformer	$\text{FastText}_{\text{fine-tuned}}$	128	6.80	12.45
Transformer	$\text{FastText}_{\text{fine-tuned}}$	256	**6.93**	**12.15**

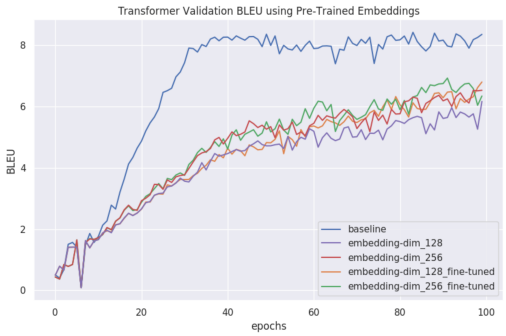


Figure 4: Transformer validation BLEU using FastText embeddings with dimensions of 128 and 256. Fine-tuned embeddings are unfrozen at the 50^{th} epoch.

5.3 Back Translation

We try to improve our transformer baseline results further by using back-translation. Table 3 shows the BLEU score for different ratio of generated examples. A ratio of 1 means that there is as much generated examples as aligned examples as per equation 1.

Table 3: BLEU Score using Back Translation

Model	Generated Ratio	BLEU	Loss
Transformer	0.5	9.95	11.85
Transformer	1	10.60	11.73
Transformer	2	11.47	11.73
Transformer	4	**11.92**	**11.29**

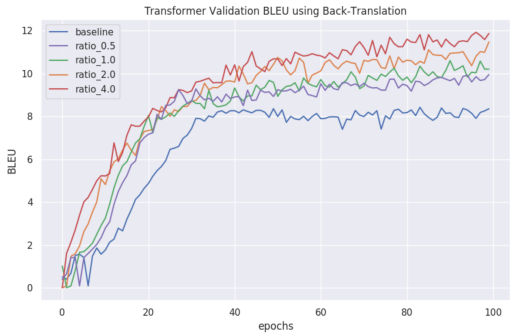


Figure 5: Transformer validation BLEU using different back-translation ratio.

5.4 Beam Search

We try to improve the results of the best performing model — the transformer trained with back-translation — by using beam search with a width (k) of 5 and length normalization parameter (α) = 0.7 and observe an improvement of 1.9 on the BLEU metric on the validation set.

Table 4: BLEU Score with and without Beam Search

Model	BLEU
Transformer	11.92
$\text{Transformer}_{\text{beam search}}$	**13.83**

5.5 Discussion

Baselines
Our two baselines perform very differently. The encoder-decoder GRU works best the larger the hidden units and embedding dimensions and the shallower the number of layers. One potential reason for this is that large hidden units and embedding dimensions could not be tried with more layers due to GPU memory constraints. This may be key to improve the performance of this model, which was very poor compared to the transformer. A shallower architecture also performs better for the transformer. Two layers perform the best but one layer has a very close BLEU score. Trying to cut all other hyperparameters by half did not result in a high drop in performance. This suggests that there might exist a smaller model with similar or better performance. For the purpose of this project, we did not search the hyperparameters in more depth but that could improve the results further. One reason that might explain the better performance of the transformer is that because of the its parallelism, we could run more epochs than the encoder-decoder in the same amount of time.

Pre-trained Embeddings
Leveraging the source monolingual data through the use of pre-trained word embeddings did not result in better translation performance. The BLEU curves of figure 2 suggest that more time would have yielded better results, although it's unclear if it could beat the baseline. Higher dimension embeddings perform slightly better. There is a 0.37 BLEU gain when doubling the embedding dimensions from 128 to 256. Fine-tuning yields a similar improvement of 0.4 BLEU for the 256 dimensions embedding. A limitation of our approach is that the hyperparameter search was done

without the FastText embeddings, so the architecture is not optimized for them. The results could be improved by training the model for more than 100 epochs and by doing a hyperparameter search with the FastText embeddings.

Back-Translation
Taking advantage of the large amount of target monolingual data through the use of back-translation improved translation performance. Using a larger ratio of synthetic examples to original examples yielded better results. Using back-translation with a generated ratio of 4 improved the baseline BLEU score by 3.5. Since a ratio of 8 could not run for 100 epochs in the 12 hours time limit of our cluster, we limited our experiments to a ratio of 4. However, as we can see in figure 4, the BLEU curves suggest that training with a higher ratio than 4 and for longer than 100 epochs could improve performance further. The fact that a higher ratio of examples with a synthetic source continues to improve results indicate that the model probably improve its translation by getting better at modeling the target language. In future work, it would be interesting to benchmark back-translation performance with a language model only trained on the target language data.

Beam Search
The beam search algorithm by itself, tends to favor shorter sentences as the joint likelihood deteriorates when we introduce more words in a sentence. To combat this we use length normalization with the parameter α set at 0.7. For further improvements, it was critical to introduce a parameter $bonus$ set to 1. We add this $bonus$ to the joint log probabilities of predictions that end with the $< eos >$ token. This essentially compels us to pick up complete sentences when we search the graph spanned by beam search for the sentence that maximizes the joint likelihood. The $bonus$ parameter was introduced in response to empirical evidence where we find beam search returning truncated shorter sentences and under-performing greedy search.
On comparing sentences generated using greedy and beam search, we see that beam search helps preserve long range context which often gets lost in greedy search. An example is shown in figure 6.

Target: *Comme nous le savons , la disparition des disparités régionales constitue un des objectifs fondamentaux de l' ue .*
Greedy Search: *Comme nous le savons la réduction des émissions de conum , il est essentiel de réduire les émissions de conum .*
Beam Search: *Comme nous le savons tous , la réduction des disparités régionales est un des objectifs fondamentaux de l' ue .*

Figure 6: Sample sentence generated with beam search versus greedy search.

Figure 8 shows the attention weights for one of the heads of the transformer trained with back-translation for a sample sentence. As we can see, the attention pair words related to each other such as "him" and "lui" or "advised" and "recommandé". Figure 7 shows the validation BLEU score as a function of the sentence length for greedy and beam search. Each point on the graph corresponds to an average of 80 examples. This plot confirms that the performance deteriorates as the sentence length increases with beam search constantly outperforming the baseline. We also find that the generated sentences are qualitatively good at first but get progressively worst the longer the sequence. Figure 11 shows examples of successful translations and figure 12 shows examples of failed translations.

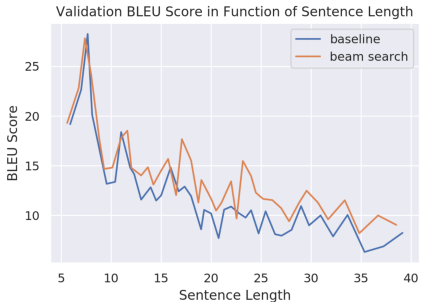


Figure 7: BLEU score as a function of the sentence length.

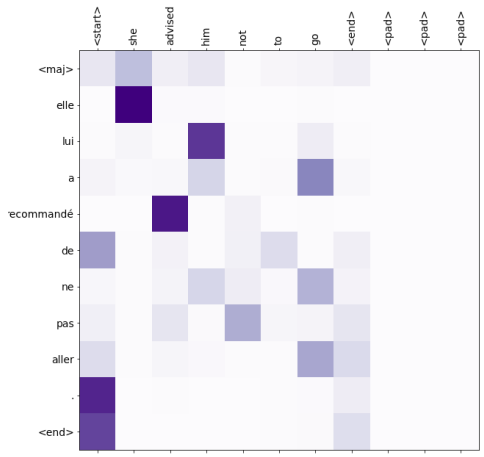


Figure 8: Attention weights of the transformer for a sentence sample.

6 Conclusion

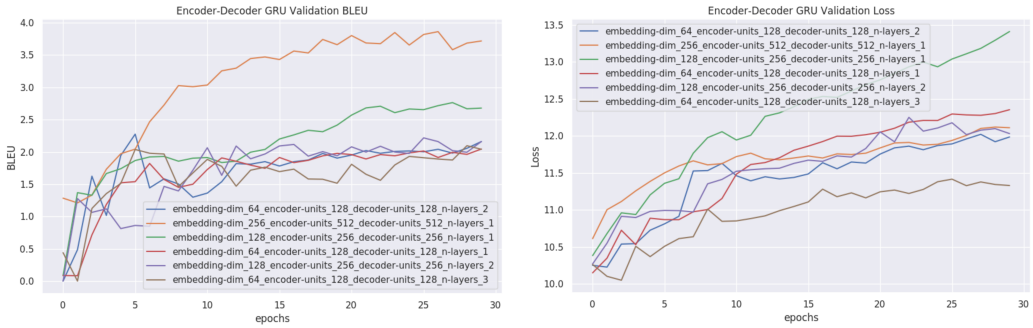
In this work, we explored two strategies to improve machine translation from monolingual data in a low resource scenario. We first do a model search to find a good baseline. The transformer model is selected with a BLEU score of 8.42. We then try to leverage source monolingual data through the use of pre-trained embeddings. We find that this strategy does not yield an improvement over the baseline. However, one limitation is that we do not search for an optimal architecture with the embeddings as input. We recommend trying to do a hyperparameter search and training the model for longer. The second strategy we tried was to use the target monolingual data through back-translation. This approach increases our baseline BLEU score by 3.5. We find that the more synthetic source examples we use, the better the performance improvement. We stop at a ratio of synthetic to original examples of 4 due to computational reason, but we suggest exploring a higher ratio in future work. Using beam search with added constraints helps us improve long range context in translations and boost BLEU score by 1.9. We also find that our translation model can infer punctuation in the target language, without the presence of punctuation in the source language.

The results suggest that good modeling of the target language is more important than the source language. In a low resource setting, having a large monolingual corpus in the target language is essential to improving results. Because of the hyperparameter search and the numerous experiments, results on the validation set might be overly optimistic. Therefore, the official performance of our approach will be computed on a held-out test set.

7 Annex

Table 5: Encoder-Decoder Hyperparameters Search

Run #	GRU Cell	Embedding Dim	Hidden Units	Layers	BLEU	Loss
1	GRU	64	128	1	2.04	12.35
2	GRU	128	256	1	2.76	13.18
3	GRU	256	512	1	**3.86**	**12.01**
4	GRU	64	128	2	2.21	10.83
5	GRU	128	256	2	2.13	11.64
6	GRU	64	128	3	2.15	10.50

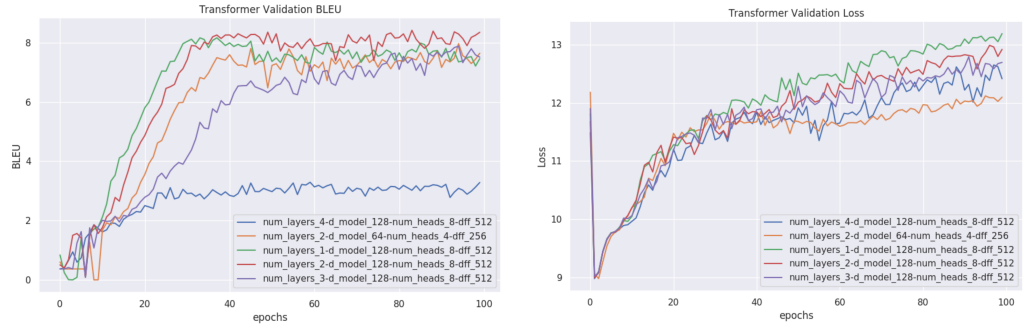


(a) (b)

Figure 9: Encoder-decoder GRU validation BLEU (a) and validation loss (b) for the different runs of the hyperparameters search.

Table 6: Transformer Hyperparameters Search

Run #	Hidden Units	d_{model}	Att. Heads	Layers	BLEU	Loss
1	512	128	8	4	3.29	**11.64**
2	512	128	8	3	7.89	12.45
3	512	128	8	2	**8.42**	12.67
4	512	128	8	1	8.17	12.01
5	256	64	4	2	7.97	11.99



(a) (b)

Figure 10: Transformer validation BLEU (a) and validation loss (b) for the different runs of the hyperparameters search.

Source: *the german presidency has also mentioned such a possibility*
Prediction: *La présidence allemande a également mentionné une telle possibilité .*
Target: *C' est un sujet qui a aussi été évoqué par la présidence allemande .*

Source: *her communication skills could be improved*
Prediction: *Sa communication pourrait être améliorée .*
Target: *Son aptitude à communiquer pourrait être améliorée .*

Source: *i do n't speak swedish*
Prediction: *Je ne parle pas suédois .*
Target: *Je ne parle pas le suédois .*

Figure 11: Examples of successful translations.

Source: *too many sweets make you fat*
Prediction: *Trop de sang , vous avez raison .*
Target: *Trop de sucreries font grossir .*

Source: *last night someone broke into the small shop near my house*
Prediction: *La nuit dernière , j' ai commencé à me concentrer sur mon petit déjeuner .*
Target: *La nuit dernière , quelqu' un a cambriolé une boutique près de chez moi .*

Source: *she asked him out on a date*
Prediction: *Elle lui a demandé un date .*
Target: *Elle lui demanda de sortir avec lui .*

Figure 12: Examples of unsuccessful translations.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *arXiv e-prints* abs/1409.0473 (Sept. 2014). URL: https://arxiv.org/abs/1409.0473.

[2] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://www.aclweb.org/anthology/D14-1179.

[3] Ankush Garg and Mayank Agarwal. *Machine Translation: A Literature Review*. Dec. 2018. URL: https://arxiv.org/pdf/1901.01122.pdf.

[4] Tom Kocmi and Ondřej Bojar. "Trivial Transfer Learning for Low-Resource Neural Machine Translation". In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 244–252. DOI: 10.18653/v1/W18-6325. URL: https://www.aclweb.org/anthology/W18-6325.

[5] Surafel Melaku Lakew et al. *A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation*. Aug. 2018. URL: https://arxiv.org/pdf/1806.06957.pdf.

[6] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *NIPS*. Curran Associates, Inc., 2013, pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[7] Jan Niehues and Eunah Cho. "Exploiting Linguistic Resources for Neural Machine Translation Using Multi-task Learning". In: *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 80–89. URL: https://arxiv.org/pdf/1708.00993.pdf.

[8] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://www.aclweb.org/anthology/P02-1040.

[9] Victor Sanh, Thomas Wolf, and Sebastian Ruder. *A Hierarchical Multi-task Approach for Learning Embeddings from Semantic Tasks*. Nov. 2018. URL: https://arxiv.org/pdf/1811.06031.pdf.

[10] Rico Sennrich, Barry Haddow, and Alexandra Birch. *Improving Neural Machine Translation Models with Monolingual Data*. 2015. arXiv: 1511.06709 [cs.CL]. URL: https://arxiv.org/pdf/1511.06709.pdf.

[11] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112. URL: https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

[12] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[13] Wikipedia contributors. *Machine translation Wikipedia, The Free Encyclopedia*. [Online; accessed 03-April-2020]. 2020. URL: https://en.wikipedia.org/wiki/Machine_translation.

[14] Wikipedia contributors. *Multi-task learning Wikipedia, The Free Encyclopedia*. [Online; accessed 05-April-2020]. 2020. URL: https://en.wikipedia.org/wiki/Multi-task_learning.

[15] Yonghui Wu et al. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016. arXiv: 1609.08144 [cs.CL].

[16] Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. “Adaptive Knowledge Sharing in Multi-Task Learning: Improving Low-Resource Neural Machine Translation”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 656–661. URL: http://users.monash.edu.au/~gholamrh/publications/acl2018-amtl.pdf.

[17] Poorya Zaremoodi and Gholamreza Haffari. “Neural Machine Translation for Bilingually Scarce Scenarios: a Deep Multi-Task Learning Approach”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1356–1365. URL: https://arxiv.org/pdf/2001.03294.pdf.

[18] Barret Zoph et al. “Transfer Learning for Low-Resource Neural Machine Translation”. In: Jan. 2016, pp. 1568–1575. DOI: 10.18653/v1/D16-1163. URL: https://arxiv.org/pdf/1604.02201.pdf.