Low-Resource Translation as Language Modeling

Tucker Berckmann and Berkan Hiziroglu
Department of Computer Science
Brown University
tucker_berckmann@brown.edu
berkan_hiziroglu@alumni.brown.edu

Abstract

We present our submission to the very low resource supervised machine translation task at the Fifth Conference on Machine Translation. The goal of this task is to create a system which translates between German and the low-resource language Upper Sorbian. We use a decoder-only transformer architecture and formulate the translation task as language modeling. To address the low-resource aspect of the problem, we pretrain over a similar language parallel corpus. Then, we employ an intermediate back-translation step before fine-tuning. Finally, we present an analysis of the system's performance.

1 Introduction

This work describes our system for translating in both directions between German (DE) and the low-resource language Upper Sorbian (HSB). German is a widespread language with tens of millions of speakers; Upper Sorbian is a West Slavic language spoken in Germany, and it is recognized as an endangered language by UNESCO (Moseley, 2010).

This system constitutes our submission to the shared task on very-low-resource supervised machine translation at WMT20.[1] The ultimate goal of the task is to translate a blind test set from Upper Sorbian into German and *vice versa*. The task is constrained, meaning that all data sets used for training are selected from a set of corpora provided by the organizers.

Our primary contribution is our application of a decoder-only language-modeling architecture to a low-resource translation task, which to our knowledge is not well-investigated.

In Sections 2 and 3, we discuss related work and our system itself. Sections 4, 5, and 6 describe our architecture. Sections 7 and 8 contain our results and analysis.

[1] http://www.statmt.org/wmt20

2 Related Work

Current approaches to machine translation include neural networks based on encoder-decoder transformers (Vaswani et al., 2017) and sequence-to-sequence models using recurrent networks (Chen et al., 2018). In both of these methods, the system learns how to produce an intermediate representation of a text sequence as a basis for the output translation. Language-neutral representations have been explored more deeply in the context of mBert (Libovický et al., 2019).

In the case of low-resource languages, where there is an absence of adequately sized parallel corpora, recent techniques focus on transfer learning (Zoph et al., 2016), relying on monolingual corpora (Lample et al., 2018), enriching the input to the system (Irvine and Callison-Burch, 2013), or expanding it through back-translation (Sennrich et al., 2016).

Techniques related to back-translation include pseudo-labeling and self-labeling. Pseudo-labeling uses partially accurate data for training (Ratner et al., 2017) generated from knowledge bases, heuristic functions and crowdsourcing. Self-labeling is an area that lies between self-supervised learning and pseudo-labeling (Caron et al., 2018; Asano et al., 2020). The model is used to predict labels for an unlabeled dataset and then is trained on this dataset.

3 Overview

Our system uses a transformer architecture, though instead of the traditional encoder-decoder layout, we use a single decoder-only transformer as do Radford et al. (2018), formulating the translation task as a language modeling task. This architecture was suggested by Radford et al. (2019) and explored concretely by Guo et al. (2019) for widely-used languages. Unlike previous approaches, in this method

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1079–1083
Online, November 19–20, 2020.

there is no intermediate representation of the input; instead, the translation is predicted directly through the attention mechanism.

Furthermore, in our submission, we rely on a similar-language pretraining task with a shared vocabulary, using Czech (CS) / English (EN) sentence pairs, similarly to Kocmi and Bojar (2018) and Nguyen and Chiang (2017).

Finally, we supplement these techniques with traditional back-translation, using monolingual corpora in the target languages.

4 Data Preprocessing

4.1 Datasets

In this work, we only use datasets that were made available by WMT20:

- HSB/DE parallel corpus (60K pairs)
- Monolingual HSB data (600K sentences)
- Monolingual DE news data (600K sentence subset)
- CS/EN parallel news corpus (60M pairs)

For the initial pretraining, we use CS/EN parallel data. The unlabeled and labeled HSB/DE parallel data are used for the back-translation and fine-tuning steps.

In Section 8, we also show a comparison to a reference pretraining dataset: a CS/DE parallel corpus (1.6M pairs).

4.2 Preprocessing Method

Figure 1 shows the preprocessing of the training corpus. This method of preprocessing the corpus allows us to use a single decoder-only transformer and train it on a classical language modeling task.

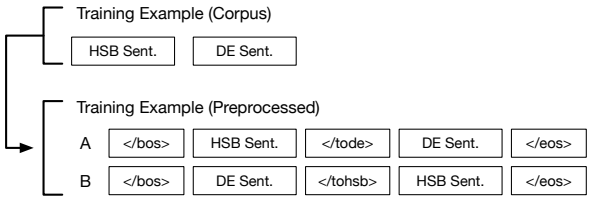

Figure 1: Preprocessing of the training corpus. The source and translation texts are concatenated with translation direction and beginning- and end-of-sequence tokens.

Since we use a CS/EN corpus on the initial pretraining step and HSB/DE corpora on the remaining steps, we create a joint byte-pair encoding which is generated by combining all of the corpora.

5 Training Method

Figure 2 shows the training method. In total, our method consists of five individual steps. These include: a pretraining step, an intermediate step made up of three sub-steps (a pre-fine-tuning step, back-translation, back-translated training), and a final fine-tuning step. The following subsections describe these steps in detail.

All of these steps (except the back-translation step, which is performed in inference mode) are performed as translation tasks using parallel corpora. The parallel corpora are either real or synthetic (in the case of the corpora resulting from back-translation).

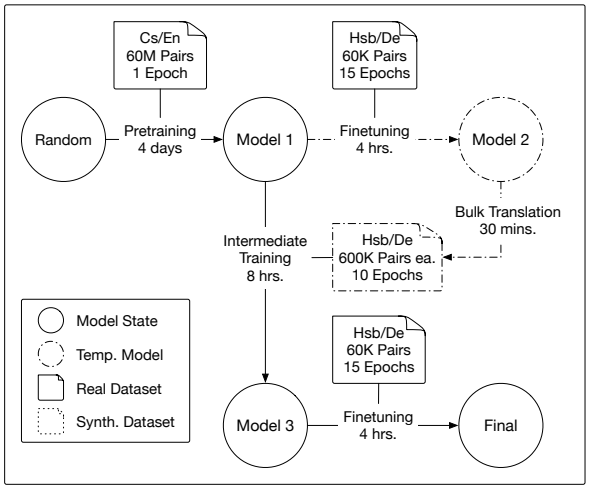

Figure 2: Training method: details are in Section 5. The dataset includes: CS/EN parallel (60M), HSB/DE parallel (60K), HSB/DE monolingal (600K each).

5.1 Initial Pretraining and Back-translation

We start by pretraining the model on a language translation task using a large (60M pair) parallel corpus consisting of Czech and English. As described by Kocmi and Bojar (2018), large (10M pair or above) parallel pretraining corpora provide significant performance gains. This is reinforced by our findings in Section 8.3.

Also, Czech and English are related to the the target languages, which can provide an additional performance benefit, according to Nguyen and Chiang (2017).

Figure 2 shows the pretraining on a CS/EN translation dataset and the first round of fine-tuning on the labeled data for the HSB/DE translation tasks.

In our method, we use the notion of *Model states*. After the pretraining step, the model reaches the *Model 1* state as depicted in Figure 2. At this step,

the model is fine-tuned and reaches the state *Model 2*. This state is used to back-translate the monolingual HSB/DE data into parallel corpora.

5.2 Back-Translated Training and Final Fine-Tuning

Output of the bulk translation is then saved and the *Model 2* state is discarded. The bulk translation is used as a parallel corpus for the back-translated training step beginning at the *Model 1* state: it consists of 600K pairs (per target language), whereby one sentence in the pair is from the monolingual corpus, and the other, parallel sentence is from the bulk translation output.

After the back-translated training, the model enters the *Model 3* state: this is the final state before the last fine-tuning. The last step is training the model in a supervised fashion on the labeled dataset.

One should note that, at this point, the model has not seen the labeled dataset yet. The state *Model 2* was trained on the labeled dataset but it is only used for the back-translation and discarded later. The final step trains the model using the highest-quality dataset: human-generated translations from the source language to the target language.

6 Implementation

We follow the GPT2 paper (Radford et al., 2019) for the model architecture, excepting hyperparameters. An overview of the system hyperparameters is shown in Table 1. We use layer normalization, a

Table 1: System Hyperparmeters

Hyperparameter	Value
Layers	4
Embedding Size	768
Attention Heads	12

standard dropout rate of 10%, and a learning rate of 5e-5 for all tasks. For the fine-tuning task, we employ L2 regularization. We train separate models for each translation direction. The total size of the model is 40M parameters.

Our choice of hyperparameters is based on a modified grid search over the attention heads, learning rate, layer count, and embedding size.

We used a single Nvidia GTX 1080TI GPU during training, and training times are shown in Figure 2. We argue that our method is time and resource efficient, easy to reproduce, and powerful.

7 Evaluation

In this section, we provide details about our implementation and the final results of the submitted system on the shared task: translation between German and Upper Sorbian.

7.1 Inference Versus Training

During the training tasks, we combine the source text, the target text, and control tokens. To use the resulting model to perform a translation of unfamiliar text, we use a slightly modified preprocessing step: we concatenate only the source text with a translation token. Since the model is trained to perform a classical language modeling task, it begins predicting the next token probabilities of the target text. We then apply beam search (with a beam width of five) to these tokens to arrive at the final translation.

7.2 Results

Table 2 shows the official BLEU score of our method on the blind test submission to WMT20. Submissions to the shared task ranged from 38.5 to 61.1 BLEU for DE to HSB translations and from 40.5 to 60.0 for HSB to DE translations.

In this section, all BLEU scores other than the blind test are calculated on the HSB/DE public test set and reference translations provided by WMT20.

Table 2: Our submission's results on the final blind test

Direction	BLEU
HSB-DE	46.0
DE-HSB	46.7

Table 3 shows a sample translation. The model's word choice is a slight generalization of the German reference, with correct grammar, spelling, and capitalization.

8 Analysis

8.1 Performance Breakdown

In order to understand the contribution of each training step to the final result, we performed experiments on different training sequences using the public HSB/DE test set provided by WMT20. The results of these experiments are reflected in Table 4. In the table, each step is cumulative and includes the steps above it: e.g., the "back-translation" step includes both fine-tuning and back-translation, but not pretraining.

Table 3: Sample Translation

Upper Sorbian Input
Otto Friedrich Bollnow mjenuje je tohodla tež hospodarske počinki.
Model output
Otto Friedrich Bollnow nennt sie deshalb auch wirtschaftliche Tugenden. **English** Therefore, Otto Friedrich Bollnow also names them economic virtues.
German Reference
Otto Friedrich Bollnow bezeichnet sie daher auch als wirtschaftliche Tugenden. **English** Therefore, Otto Friedrich Bollnow also describes them as economic virtues.

Table 4: Breakdown of BLEU score as training tasks are added: the results are cumulative

Step	DE-HSB	HSB-DE
Fine-tuning only	27.4	27.3
Back-translation	38.2	38.1
Pretraining	44.6	42.9
Blind test	46.7	46.0

We conclude from these experiments that the most significant gains came from back-translation (around 10 BLEU), followed by the pretraining step (4-6 BLEU).

For reference, we reiterate the blind test results in Table 2. The blind test results are different from the pretraining step due to differences in the data set.

8.2 Pretraining Task Selection

We considered using unsupervised learning as a pretraining task; however, a comparison of unsupervised pretraining in the target language with translation-task pretraining using related languages showed that the translation task had a greater impact on the final model's performance.

In this experiment, we compared the effect of different pretraining tasks on the model's translation performance. Recall that our architecture formulates the translation task as a language modeling task. Since the architecture acts as a language model, it is also possible to pretrain the model, without modification, on unsupervised text in the target languages.

To compare the unsupervised language modeling pretraining task with a translation pretraining task, we pretrained one model with the full HSB/DE unsupervised data set (600K sentences each, 10 epochs), a second model with a CS/DE parallel corpus (1.6M pairs, 3 epochs), and a third model with a subset of the CS/EN parallel corpus (60M pairs, 0.17 epochs), and then fine-tuned each of them using the supervised data set.

We compare these models to a baseline (fine-tuned only) model in Table 5. From these results, we conclude that the similar language translation tasks are more effective pretraining tasks than unsupervised language modeling in this context. The two related-language pretraining tasks were comparable in performance, though we only used a fraction of the CS/EN corpus due to its much larger size.

Table 5: Target-task BLEU score after fine-tuning, given pretraining tasks in various languages

Pretraining Task	Type	DE -HSB	HSB -DE
None	-	27.4	27.3
Unsupervised	HSB/DE	28.1	29.5
Translation	CS/DE	31.7	31.4
Translation	CS/EN	31.5	32.7

8.3 Pretraining Corpus Size

Finally, we examined the effect of the number of pretraining epochs on the final BLEU score. As shown in Table 6, roughly doubling the corpus size led to an increase of nearly 1.0 BLEU in the final model performance. This represents close to 20% of the performance increase we attribute to our pretraining task, which suggests that an even larger corpus, or additional pretraining epochs, would contribute further to model performance.

Table 6: Effect of 60M-pair pretraining corpus size (in epochs) on final HSB->DE BLEU score

Epochs	BLEU Score
0.42	41.4
1.00	42.3

9 Conclusion

Since our model produces high-quality translations, we have shown that a small decoder-only transformer, configured to perform classical language modeling, is an effective translation system for low-resource language pairs. Furthermore, we have shown that a similar language translation pretraining task can contribute substantially to the quality of such translation systems. Finally, we have provided an analysis of the model's components and their relative contribution to its ultimate performance.

Further investigation would be needed to understand our model's relationship to other architectures under the same data sets and pretraining tasks.

Acknowledgments

We would like to thank Prof. Eugene Charniak for the helpful discussion. We would also like to thank Prof. Ellie Pavlick for her help in reviewing the paper and the method.

References

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Michael Schuster, Zhi-Feng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *ACL*.

Yinpeng Guo, Yi Liao, Xin Jiang, Qing Zhang, Yibo Zhang, and Qun Liu. 2019. Zero-shot paraphrase generation with multilingual language models. *arXiv preprint arXiv:1911.03597*.

Ann Irvine and Chris Callison-Burch. 2013. Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 262–270.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Christopher Moseley, editor. 2010. *Atlas of the World's Languages in Danger*. Memory of peoples Series. UNESCO Publishing.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access.

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

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

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.