The LMU Munich System for the WMT20 Very Low Resource Supervised MT Task

Jindřich Libovický and **Viktor Hangya** and **Helmut Schmid** and **Alexander Fraser**
Center for Information and Language Processing
LMU Munich
{libovicky,hangya,schmid,fraser}@cis.lmu.de

Abstract

We present our systems for the WMT20 Very Low Resource MT Task for translation between German and Upper Sorbian. For training our systems, we generate synthetic data by both back- and forward-translation. Additionally, we enrich the training data with German-Czech translated from Czech to Upper Sorbian by an unsupervised statistical MT system incorporating orthographically similar word pairs and transliterations of OOV words. Our best translation system between German and Sorbian is based on transfer learning from a Czech-German system and scores 12 to 13 BLEU higher than a baseline system built using the available parallel data only.

1 Introduction

In this paper, we describe systems for translation between German and Upper Sorbian developed at LMU Munich for the WMT20 shared task on very low-resource supervised MT.

Upper Sorbian is a minority language spoken by around 30,000 people in today's German state of Saxony. With such a small number of speakers, machine translation and automatic processing of Sorbian is an inherently low-resource problem without any chance that the resources available for Sorbian would ever approach the size of resources for languages spoken by millions of people. On the other hand, being a Western Slavic language related to Czech and Polish, it is possible to take advantage of relatively rich resources collected for these two languages.

The German-Sorbian systems presented in this paper are neural machine translation (NMT) systems based on the Transformer architecture (Vaswani et al., 2017). We experiment with various data preparation and augmentation techniques: back-translation (Sennrich et al., 2016b), finetuning systems trained for translation between Czech and German (Kocmi and Bojar, 2018), and data augmentation by including German-Czech parallel data with the Czech side translated to Upper Sorbian by an unsupervised system that includes an unsupervised transliteration model for guessing how to translate out-of-vocabulary Czech words to Upper Sorbian.

Our experiments show the importance of data augmentation via stochastic pre-processing and synthetic data generation. The best systems were trained by transfer-learning from a Czech-German system. However, compared to data augmentation, transfer learning from Czech-German translation only produces a minor improvement. Based on the preliminary shared task results, the presented systems scored on the 4th place among 10 competing teams in the shared task.

2 Related Work

Until recently, phrase-based approaches were believed to be more suitable for low-resource translation. Koehn and Knowles (2017) claimed that a parallel dataset of at least 10^7 tokens is required for NMT to outperform phrase-based MT. This view was also supported by the results of Artetxe et al. (2018b) and Lample et al. (2018), who showed that phrase-based approaches work well for unsupervised MT, at least in the early stages of the iterative back-translation procedure.

Recently, Sennrich and Zhang (2019) revisited the claims about data needs of supervised NMT and showed that with recent innovations in neural network and careful hyper-parameter tuning, NMT models outperform their phrase-based counterparts with training data as small as 100k tokens (15 times smaller than the data provided for this shared task).

Standard techniques for low-resource machine translation include data augmentation with rule-based substitutions (Fadaee et al., 2017), by sam-

Data		# sent.	# tok.	$\frac{\text{\# tok.}}{\text{\# sent.}}$
Train	de	60k	822k	13.7
	hsb		738k	12.3
Devel	de	2k	28k	13.8
	hsb		25k	12.5
Devel test	de	2k	28k	13.9
	hsb		25k	12.7
German-Czech newstest2019	de	2k	49k	24.5
	cs		43k	22.0
German-Czech parallel	de	14.7M	234M	15.9
	cs		219M	14.8

Table 1: Statistics on the parallel data compared to German-Czech News Test 2019 and parallel German-Czech data (see Section 3.3).

pling synthetic noise (Wang et al., 2018; Provilkov et al., 2020), or by iterative back-translation (Hoang et al., 2018). Another class of approaches relies on transfer learning from models trained for high-resource language pairs of more or less similar languages (Zoph et al., 2016; Nguyen and Chiang, 2017; Kocmi and Bojar, 2018).

3 Data

We used several types of data to train our systems. The organizers provided authentic parallel data and Sorbian monolingual data. We also use German and Czech News Crawl data and Czech-German parallel data available in Opus (Tiedemann, 2012).

3.1 Authentic Parallel Data

The organizers of the shared task provided a parallel corpus of 60k sentences, and validation and development test data of 2k sentences each.

The basic statistics about the data are presented in Table 1. Note that the sentences are on average much shorter and therefore also likely to be structurally simpler than in the type of sentences usually used in the WMT test sets.

3.2 Monolingual Data

In total 696k monolingual Sorbian sentences were provided by the organizers. We noticed that the monolingual Sorbian data contain many OCR-related errors originating from hyphenation. We thus removed all sentences ending with a hyphen. Additionally, we merged tokens ending with a hyphen with the adjacent one if such merging results in a known Sorbian word. This filtered out 1.6k sentences and did 12k token merges.

The monolingual Sorbian data were used for training the unsupervised Czech-Sorbian translation system (see Section 4.1) and for back-translation in Sorbian-German systems.

Besides, we use 60M German and 60M Czech sentences from the NewsCrawl data provided as monolingual data for WMT shared tasks (Barrault et al., 2019). The monolingual data were used for generating synthetic training data via back- and forward-translation both for the German-Sorbian and German-Czech systems. In addition, the Czech monolingual data was used in the unsupervised Czech-Sorbian translation system as well.

3.3 German-Czech Data

For transfer learning and the creation of synthetic data, we also used German-Czech parallel data. We downloaded all available parallel datasets from the Opus project (Tiedemann, 2012), which gave us 20.8M parallel sentences, which we further filtered.

First, we filtered the parallel sentences by length. We estimated the mean and the standard deviation of the length ratio of German and Czech sentences and kept only those sentence pairs whose length ratio fitted into the interval of two times standard deviation around the mean. Then, we applied a language identifier from FastText (Grave et al., 2018) and only kept sentence pairs identified as German-Czech. The filtering lefts us with 14.7M parallel sentences.

4 Synthetic data from Czech-German

Since Upper Sorbian is related to Czech, we generate additional synthetic parallel German-Sorbian data by translating the Czech side of the German-Czech parallel data. For this, we use an unsupervised statistical MT system which includes mined Czech-Sorbian transliteration word pairs for better performance.

4.1 Unsupervised SMT

We follow the approach of Artetxe et al. (2018b) to build an Unsupervised Statistical Machine Translation (SMT) system. In the following description, we mainly focus on the steps that we changed compared to the original system and keep the description of the other steps brief.

In the first step, we build 300-dimensional monolingual n-gram embeddings for both Czech and Sor-

bian using *FastText skip-gram* (Bojanowski et al., 2017) on the above mentioned monolingual data. We restrict the vocabulary to the most frequent 200k, 400k, and 400k 1-, 2- and 3-grams, respectively. We map these embeddings to a shared bilingual space using *VecMap* (Artetxe et al., 2018a). In contrast to the original unsupervised SMT pipeline, which builds bilingual word embeddings (BWEs) without any cross-lingual signal, we use identical words occurring in both languages as the seed lexicon for the mapping. We found that the available small monolingual Sorbian corpus is not adequate to build BWEs in a fully unsupervised way. The corpora are tokenized and true-cased using *Moses* tools (Koehn et al., 2007). We note that because there are no available language rules for Sorbian, we used Czech rules for tokenization, which is reasonable because of the similarity of the two languages.

We build phrase tables for both translation directions. For each source n-gram, we take 100 candidates with the closest embeddings based on cosine similarity and additional 100 candidates with the smallest edit distance. We calculate 5 scores for each pair: phrase and lexical translation probabilities and their inverse as in (Artetxe et al., 2018b), and their normalized edit distance. For phrases, the latter is calculated by pairing each source word with the most similar target side word and taking the average edit distance of each of these pairs as the normalization constant. In addition to the phrase tables, we train language models using the monolingual corpora.

We use the validation set from the shared task (with the German side machine-translated to Czech) to tune the parameters with MERT instead of tuning on synthetic data. Finally, we run 3 iterative refinement steps.

4.2 Translating OOVs by Transliteration

Because of the small monolingual data, the Sorbian vocabulary is relatively small. To improve on this problem, we exploit the similarity of Upper Sorbian and Czech by translating Czech out-of-vocabulary (OOV) words to Upper Sorbian, using transliteration. More precisely, we transliterate Czech words from the German-Czech parallel data which were not seen by the SMT system during training, assuming that the translations of these words are missing in the Sorbian vocabulary on the target side as well. We extracted the training data for the transliteration system using a preliminary *transliteration mining* model, *filtered* the data using a preliminary transliteration model, and trained the final *transliteration model* on the filtered data.

Transliteration mining. Our transliteration mining is similar to the model by Sajjad et al. (2012). It consists of a transliteration submodel and a noise submodel.

The *transliteration submodel* is a unigram model over transliteration units (TUs) which jointly generates a source and a target language string. The English-German transliteration pair *(Gorbatchev, Gorbatschow)* could be generated as the following sequence of TUs: G:G o:o r:r b:b a:a t:t s: c:c h:h e:o v:w. We use only 1-1, 0-1, and 1-0 TUs. The probability $\mathbf{p}(\mathbf{a})$ of a sequence of TUs is the product of the unigram probabilities $p(a_i)$:

$$p(\mathbf{a}) = p(a_1, ..., a_n) = \prod_{i=1}^{n} p(a_i)$$

Probability $p_{\text{trans}}(\mathbf{s}, \mathbf{t})$ of a string pair is obtained by summing over all possible alignments $\mathbf{a}$:

$$p_{\text{trans}}(\mathbf{s}, \mathbf{t}) = \sum_{\mathbf{a} \in \text{align}(\mathbf{s}, \mathbf{t})} p(\mathbf{a})$$

The *noise submodel* independently generates a source string $\mathbf{s}$ and a target string $\mathbf{t}$ using two unigram models over the source and the target language characters, respectively. The probability of a string pair is the product of the two monolingual string probabilities:

$$p_{\text{noise}}(\mathbf{s}, \mathbf{t}) = p_{\text{src}}(\mathbf{s})\ p_{\text{tgt}}(\mathbf{t})$$

The monolingual probability of the source string (and analogously the target string) is defined as a product of letter unigram probabilities.

Sajjad et al. (2012) *interpolate* the noise model and the target model as a linear combination. Unfortunately, such a model also extracts near-transliterations which differ from a true transliteration by e.g., an inflexional affix, such as *(Gorbatchev, Gorbatschows)*.

Instead, we combine the two submodels by *concatenation*. Our model produces a word pair by (i) generating two word prefixes[1] $\mathbf{s}^p$ and $\mathbf{t}^p$ using the noise model, (ii) generating two middle parts $\mathbf{s}^m$ and $\mathbf{t}^m$ using the transliteration model, and

[1] Here, the terms *prefix* and *suffix* are not used in a linguistic sense.

(iii) generating two suffixes $\mathbf{s}^s$ and $\mathbf{t}^s$ using the noise model. The intuition is that if the most probable way to generate a pair does not use prefixes or suffixes, it is a transliteration. Here the non-transliteration pair *(Gorbatchev, Gorbatschows* might be most probably obtained by generating two empty strings as prefixes with the noise submodel, the TU sequence G:G o:o r:r b:b a:a t:t s: c:c h:h e:o v:w with the transliteration submodel, and an empty suffix and the suffix *s* with the noise model.

The probability is defined as follows:

$$p(\mathbf{s^p}, \mathbf{s^m}, \mathbf{s^s}, \mathbf{t^p}, \mathbf{t^m}, \mathbf{t^s}) = p_{\text{noise}}(\mathbf{s^p}, \mathbf{t^p}) p_{\text{trans}}(\mathbf{s^m}, \mathbf{t^m}) p_{\text{noise}}(\mathbf{s^s}, \mathbf{t^s})$$

The total probability of a word pair is obtained by summing over all possible splits:

$$p(\mathbf{s}, \mathbf{t}) = \sum_{\substack{\mathbf{s^p}, \mathbf{s^m}, \mathbf{s^s}, \mathbf{t^p}, \mathbf{t^m}, \mathbf{t^s} \\ \in \text{split}(\mathbf{s}, \mathbf{t})}} p(\mathbf{s^p}, \mathbf{s^m}, \mathbf{s^s}, \mathbf{t^p}, \mathbf{t^m}, \mathbf{t^s})$$

The parameters of the transliteration submodel are trained using the EM algorithm on the list of transliteration candidates. The parameters of the monolingual models are estimated directly from the data and kept fixed during training. After the EM training, we compute for each candidate pair, the most probable split of the two words into prefix/middle/suffix, and the most probable alignment of the two middle parts using the Viterbi algorithm. If all prefixes and suffixes are empty, the candidate pair is extracted as a probable transliteration.

We run the transliteration mining on lower-cased data and consider all possible word pairs with a reasonable edit distance. The mining process returns the extracted transliteration candidates and their most probable TU sequence, respectively.

Transliteration filtering. The mining process only relies on the unigram probabilities, which is often suboptimal. Therefore, we add a filtering step that scores each transliteration pair using an n-gram transliteration model and eliminates pairs with a low score.

We train a Kneser-Ney-smoothed trigram transliteration model on the TU sequences of the transliterations extracted using the transliteration mining model.

For each extracted transliteration pair, we compute negative log probabilities:

- L_1 of the corresponding TU sequence;
- L_2 of the best source-to-target transliteration; and
- L_3 of the best target-to-source transliteration.

We filter out a word pair if $L_2 - L_1 + L_3 - L_1 > 10$. Note that all three probabilities are joint probabilities and that the same transliteration model can be used in both directions.

Unsupervised SMT	12.0
+ edit distance	13.3
+ transliteration	13.8

Table 2: BLEU scores of the Czech-Sorbian system with gradually added techniques measured on the Upper Sorbian-German test set where the German side has been machine-translated to Czech.

Transliteration Generation. We train the final transliteration model on the TU sequences of the filtered transliteration pairs and use the model to generate Sorbian transliterations for Czech OOV words. We lowercase the Czech words before transliteration and transfer the casing from the original Czech words to their Sorbian transliterations.

Using the model, we generate transliterations for Czech words not seen by the unsupervised SMT system during training, i.e., we take all the words from the Czech side of the parallel data which are not present in the used Czech monolingual corpus. To add these word pairs to the SMT system, we consider them as a parallel corpus and concatenate it to the synthetic parallel data created in the iterative refinement steps and also update the language models. We run two additional refinement steps on top of the three mentioned in 4.1. Finally, we create the synthetic German-Sorbian data by translating the Czech side of the German-Czech data and feed it to our final NMT system, as described below.

Table 2 shows the translation quality of the unsupervised SMT system. The basic setup relies only on BWEs to build the initial phrase tables. Next, we add edit distance information, and finally, we use the mined transliteration pairs as well. However, note that the BLEU scores are very approximate because the source side of the test is machine-translated.

5 Experimental Setup

For the translation between German and Sorbian, we experimented with NMT models based on

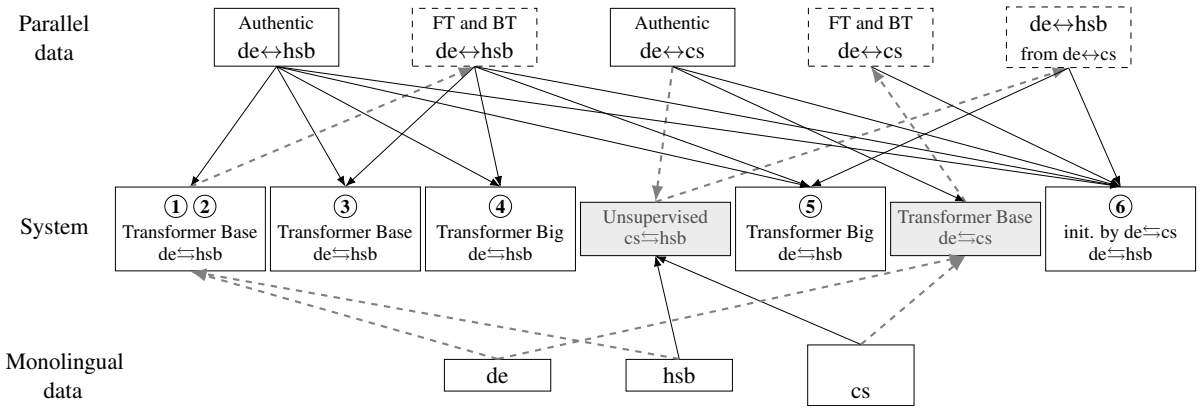

Figure 1: Overview of datasets and systems that were used to generate synthetic data. Solid arrows denote training a system, dashed gray arrows denote using the model for data generation. Synthetic datasets have dashed boxes.

Transformers (Vaswani et al., 2017). We followed known best practices for architecture and optimization choices. In our experiments, we mostly focus on data engineering.

5.1 Model Architecture and Optimization

We use the Transformer architecture (Vaswani et al., 2017) as implemented in Marian (Junczys-Dowmunt et al., 2018). For the initial experiments, we used the Base architecture (6 layers, hidden state of size 512, 8 attention heads, feed-forward layer 1024), and Big for the later experiments (12 layers, hidden state 1024, 16 attention heads, feed-forward layer 4096). We follow the default standard learning rate schedule proposed by Vaswani et al. (2017) with learning rate $3 \cdot 10^{-4}$. We use 16k warm-up steps for the Base architecture and 32k warm-up steps for the Big architecture.

The Base architecture is used for the initial systems which generate synthetic data via backward- and forward-translation. We use the Big architecture for the rest of the systems.

5.2 Training Data Preparation

An overview of the data generation and system training steps is provided in Figure 1.

We use a common BPE-based vocabulary (Sennrich et al., 2016c) for all systems which allows us to better ensemble our systems. Instead of proper tokenization, we use the pre-tokenization heuristic from SentencePiece (Kudo and Richardson, 2018) as implemented in YouTokenToMe.[2] The BPE vocabulary consists of 16k merges and was fit using the authentic parallel training data only.

[2]https://github.com/VKCOM/YouTokenToMe

We apply BPE-dropout (Provilkov et al., 2020) of 0.1 on both the source and the target side of the data. We oversample the monolingual data 1000 times and with different segmentations (Model 2). We hypothesize that in the very low-resource setup, the BPE dropout serves more as a data-augmentation technique than as regularization.

Due to hardware limitations, we limit the data mixes for training the Big architectures to 180M parallel sentences. One third of the data mix consists of oversampled authentic parallel data. In one set of experiments (Models 3, 4), the rest of the data consists of synthetic data: an equal number of samples of forward- and back-translation (which means that the monolingual Sorbian data is oversampled approximately 80×). In another set of experiments (Model 5), we additionally sample data from the machine-translated Czech-German data set where the Czech part has been automatically translated to Upper Sorbian. Following Caswell et al. (2019), we tag the synthetic data, having a separate tag for each of the synthetic data types.

Further, we experiment with finetuning models originally trained for translation between Czech and German. The data for the parent models is prepared using the same protocol as for Model 4. Following Kocmi and Bojar (2018), we train the parent model until convergence and continue training with the German-Sorbian data. Based on preliminary results, we use the data mix for Model 4 for the German-to-Sorbian translation direction and the data mix for Model 5 for translating from Sorbian into German.

Model		hsb→de		de→hsb	
1	Transformer Base, parallel only	43.4	.695	45.6	.702
2	(1) + BPE dropout	50.9	.745	51.7	.747
3	(2) + back- and forward-translation	51.6	.766	52.4	.765
4	Transformer Big, same data as (3)	53.0	.766	55.3	.765
5	(4) + synthetic data from cs-de	54.2	.766	54.9	.766
6	(4/5) initialized by cs⇄de	55.4	.772	55.9	.775
7	Ensemble 4× (4/5)	55.0	.772	55.9	.773
8	Ensemble 3× (6)	55.6	.773	56.2	.776
9	Ensemble 4× (4/5) and 3× (6)	56.0	.777	56.9	.769
10	(4/5) trained right-to-left	53.7	.765	55.1	.769
11	(9) + right-to-left rescoring	56.0	.778	57.0	.779

Table 3: BLEU scores and chrF scores (in small font) on development test data for Sorbian-to-German (hsb→de) and German-to-Sorbian (de→hsb) translations.

5.3 Model Ensembling

Following Sennrich et al. (2016a), we also experiment with ensembling several systems and combining systems trained in the left-to-right and right-to-left direction.

We trained four models from random initialization and three models by transferring from Czech-German translation. Note that the transferred models were initialized by the same model and only differed in the order of the training data.

Further, we trained two models in the right-to-left direction, starting from random initialization.

6 Results

The quantitative results in terms of BLEU score (Papineni et al., 2002) and ChrF (Popović, 2017) score are presented in Table 3. The results were measured using SacreBLEU.[3]

The Base architecture trained using the parallel data only (Model 1) reaches a surprisingly high BLEU score, which is probably due to the quality of the manually curated training data, domain closeness of the train and test data, and relatively simple sentences both in the train and test sets.

The data augmentation using BPE-dropout (Model 2) seems to have a substantial effect on the translation quality, improving the translation by 6–7 BLEU points. This is a much larger effect than Provilkov et al. (2020) reported. However, they also observed a larger positive effect on smaller datasets. Unlike Sennrich and Zhang (2019), we did not find any benefits of using a small BPE-based vocabulary or tuning learning rate. However, the positive effect of the small vocabulary might be partially emulated by the BPE dropout.

Adding the back- and forward-translated data in the training data improved the translation quality only slightly (Model 3). A large positive effect can be achieved by switching to the Big architecture (Model 4). Adding the synthetic data generated from Czech-German parallel data improve only the Sorbian-to-German translation direction (Model 5), presumably because the quality of the synthetic Sorbian side of the corpus is too low to be used as a target side.

Transfer learning from German-Czech models further improves the translation quality by approximately 1 BLEU point. These are thus the best single models we have developed and our contrastive submission to the shared task.

Additional improvements were reached by model ensembling. Ensembling both the model trained from random initialization and transfer learning models improves the translation by approx. 1 BLEU point. Ensembling these two model types together further improves the translation quality by around half BLEU point.

The model generating the translation right-to-left reach translation quality that is comparable to the left-to-right models. However, rescoring of the n-best lists generated by left-to-right ensembles by the right-to-left models improves the translation quality only negligibly. The rescored ensemble was our primary submission to the shared task.

[3] https://github.com/mjpost/sacrebleu

7 Conclusions

We presented NMT systems for translation between German and Upper Sorbian. Due to the domain closeness and relative simplicity of the test data, we were able to achieve BLEU scores over 50 using the parallel data only. The crucial component was the use of BPE-dropout for both the source and target side.

Further translation quality improvements were achieved by generating synthetic training data by back- and forward-translation. Additionally, we generated synthetic data by machine-translating the Czech side of a parallel German-Czech corpus. For that, we built an unsupervised SMT system that additionally utilizes an unsupervised transliteration system for the translation of OOV tokens.

Our best single system is based on transfer learning, i.e., initializing the model by a Czech-German system, reaching 1–2 higher BLEU scores compared to systems based on Sorbian and German data only. Further minor improvements were achieved by model ensembling and right-to-left rescoring.

Acknowledgments

The work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 640550) and by German Research Foundation (DFG; grant FR 2829/4-1).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

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, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion*

Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

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.

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

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 469–477, Jeju Island, Korea. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

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.