UdS-DFKI@WMT20: Unsupervised MT and Very Low Resource Supervised MT for German↔Upper Sorbian

Sourav Dutta[1]*, Jesujoba O. Alabi[1,3]*, Saptarashmi Bandyopadhyay[2], Dana Ruiter[1], Josef van Genabith[1,3]

[1]Saarland University, Saarbrücken, Germany
[2]University of Maryland, College Park, MD 20742
[3]DFKI GmbH, Saarbrücken, Germany
souravd@coli.uni-saarland.de
sapta.band59@gmail.com, druiter@lsv.uni-saarland.de
{jesujoba_oluwadara.alabi, josef.van_genabith}@dfki.de

Abstract

This paper describes the UdS-DFKI submission to the shared task for unsupervised machine translation (MT) and very low-resource supervised MT between German (de) and Upper Sorbian (hsb) at the Fifth Conference of Machine Translation (WMT20). We submit systems for both the supervised and unsupervised tracks. Apart from various experimental approaches like bitext mining, model pretraining, and iterative back-translation, we employ a factored machine translation approach on a small BPE vocabulary.

1 Introduction

This paper describes the UdS-DFKI submission to the unsupervised and very low resource supervised tasks of WMT20 for German to Upper Sorbian (*de→hsb*) and Upper Sorbian to German (*hsb→de*). Our submitted systems are constrained for the very low resource supervised and unconstrained for the unsupervised task, in that we use Wikipedia dumps as additional data.

Current machine translation systems that deal with low-resource languages are based on unsupervised neural machine translation, semi-supervised methods and pre-trained models leveraging monolingual data (Guzmán et al., 2019), and multilingual systems among others. In this work, we explore different systems which include baseline NMT, factored NMT (Sennrich and Haddow, 2016a), iterative backtranslation, self-supervised NMT (SSNMT) (Ruiter et al., 2019) and pretraining with XLM (Lample and Conneau, 2019) using transformer-base models (Vaswani et al., 2017) for the training of the systems.

This paper begins by presenting the data we used for the tasks and the preprocessing pipeline (Section 2). This is followed by an overview of the training setup (Section 3) and the methods we applied (Section 4). Section 5 summarizes our findings, followed by a discussion of the results in Section 6. We conclude the paper and propose some possible future work in Section 7.

2 Data

Unsupervised Task For Sorbian, we use the given data from the Sorbian Institute (Ins_{hsb}), from Witaj Sprachzentrum (Witaj_{hsb}), and web-scraped data (Web_{hsb}). Table 1 gives a summary of the data we use in the unsupervised track. We use the Europarl (EP_mono_{de}, (Koehn, 2005)) and News Commentary (NC_mono_{de}, (Barrault et al., 2019)) datasets for the monolingual German data. Apart from this, we also use Wikipedia Dumps[1] for both German and Upper Sorbian. We extract articles using Wikiextractor[2], which are aligned using Wikipedia *langlinks*[3] to create a comparable corpus for SSNMT extraction.

Supervised Task Apart from the provided parallel data, we use high-quality EUROPARL (EP, (Koehn, 2005)) and medium-quality JW300 (Agić and Vulić, 2019; Tiedemann, 2012) corpora for *de-cs-hsb*. For parallel text mining with LASER (Schwenk, 2018; Artetxe and Schwenk, 2019), we use the combination of all the monolingual corpora of German and Upper Sorbian from the unsupervised section of Table 1, which is discussed in detail later in Section 4.3.

Preprocessing Our preprocessing steps include normalization, tokenization, deduplication, and truecasing. We attach feature labels related to the source language (<src>), target language (<tgt>), and the data quality (<quality>), for

* Equal contribution

[1] https://dumps.wikimedia.org/ (March 2020)
[2] https://github.com/attardi/wikiextractor
[3] https://www.mediawiki.org/wiki/API:Langlinks

every individual sentence. The quality of a sentence depends on the corpus it is from and the quality tags of <low>, <medium>, or <high> are added to all sentences of the corpora according to the quality labels assigned to the data provided for the shared task: e.g. Ins_{hsb} is high quality Sorbian. A typical sentence from the corpus after our preprocessing pipeline has the following format:

```
<src> <tgt> <quality> sentence
```

After factoring (4.2), we proceed to apply joint byte-pair encoding (BPE) (Sennrich et al., 2016b) on the corpora to finally get our preprocessed data which we use for training all our NMT models. Unless specified otherwise, we use a default of 5k merge operations.

Corpus	# Sentences	# Tokens
Unsupervised		
Ins_{hsb}	339k	5,044k
Witaj_{hsb}	222k	2,672k
Web_{hsb}	134k	1,677k
EP_mono_{de}	2,107k	55,557k
NC_mono_{de}	422k	8,942k
Wiki_{de}	833k	36,531k
Wiki_{hsb}	76k	2,402k
Supervised		
Bitext_{de}	60k	1,002k
Bitext_{hsb}	60k	737k
EP_{de}	568k	13,098k
EP_{cs}	568k	11,571k
JW300_{de}	1,179k	20,888k
JW300_{cs}	1,179k	19,144k
Dev & Test		
Dev20_{de}	2k	24k
Dev20_{hsb}	2k	21k
DevTest20_{de}	2k	24k
DevTest20_{hsb}	2k	22k

Table 1: Statistics (in thousands) of different corpora used for the unsupervised and supervised tasks.

3 Systems

MT Systems We train all our models using the Transformer-base architecture in the OpenNMT-py (Klein et al., 2017) framework extended for SSNMT[4] (Ruiter et al., 2019). The setting for the Transformer base is the same as in Vaswani et al. (2017) with 6 encoder-decoder layers after having explored other options of Transformer depth. We set the dropout to 0.4 in all experiments. We use *adam* (Kingma and Ba, 2014) for optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.998$. The learning rate is varied from 0 to 2 with a warm update of 4000 and decayed using *noam*. Lower values of learning rate were avoided due to slower training and lower accuracy scores. We use a batch size of 50. The Phrase-Based Statistical MT systems (PBSMT) are standard *Moses* (Koehn et al., 2007) systems trained without applying BPE to the data.

Initialization The NMT models are initialised with cross-lingual word embeddings calculated on the monolingual corpora using `word2vec`[5] (Mikolov et al., 2013) (skip-gram) and unsupervised `VecMap`[6] (Artetxe et al., 2017).

Pre-trained Sentence Representations For XLM (Lample and Conneau, 2019), we pre-train and fine-tune the model using drop out of 0.1, batch size of 32 with a joint BPE of $10k$ ($10k$ showed better results for XLM), learning rate of 0.0001, and a sequence length of 265 using 512 and 1024 embedding dimensions respectively.

Evaluation Metric We use BLEU[7] (Papineni et al., 2002) scores to evaluate the performance of our trained models.

4 Techniques

4.1 Iterative Backtranslation

For the unsupervised task, we use an SSNMT system as described in Ruiter et al. (2019) to extract parallel sentences from the Wikipedia dumps. SSNMT jointly and iteratively extracts parallel data, and learns the MT task on the extracted parallel data. The resulting trained NMT model is our base model (M_0).

For iterative back-translation (Hoang et al., 2018), we take a model M_i and use it to translate the *hsb* monolingual data mono_{hsb} and EP_mono_{de} to generate mono^i_{de} and EP_mono^i_{hsb} respectively. Following Sennrich et al. (2016a), we use the generated data at iteration i on the source side with

[4] https://github.com/ruitedk6/comparableNMT

[5] https://github.com/tmikolov/word2vec

[6] https://github.com/artetxem/vecmap

[7] We use the *Moses* multi-bleu script for evaluation. https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

the original data on the target to train a new model M_{i+1}. This is done iteratively, in our case until $i = 5$.

As the translation quality of M_0 is very low, this model is replaced by a PBSMT system which is trained on the data that M_0 has extracted, in order to generate the back-translation to be used for M_1.

Model	BLEU	
	hsb→de	*de→hsb*
M_0	6.46	6.09
M_1	8.53	8.31
M_2	9.81	10.04
M_3	10.47	13.51
M_4	**11.31**	11.57
M_5	9.13	**13.61**

Table 2: BLEU scores of iterative-backtranslation models per iteration, calculated on Dev20.

The resulting BLEU scores on Dev20 for each of the iterations is shown in Table 2. The best performance for *hsb→de* is achieved at $i = 4$ (11.31 BLEU) and for *de→hsb* at $i = 5$ (13.61 BLEU). These constitute two of the models submitted to the unsupervised task.

4.2 Factorization

Limited monolingual language analysis tools and few linguistic analysis tools with acceptable performance are available for low-resource (LowRes) languages. In our experiments, we explore factored machine translation (García-Martínez et al., 2016; Sennrich and Haddow, 2016b; Koehn and Knowles, 2017). This approach can play a significant role in increasing grammatical coherence. Syntactic and semantic information can be useful to generalize neural models trained on parallel corpora.

We augment our parallel data to include factors like *lemma* (using Snowball Stemmer (Porter, 2001)) and *PoS* tags (using *spaCy*[8] open source library (Honnibal and Montani, 2017)) for German words. The language-agnostic UDPipe trainable pipeline (Straka et al., 2016) has been used for lemmatization and PoS tagging for Sorbian words. We follow an approach similar to Bandyopadhyay (2019, 2020), where we factor the data at word-level to include the root word (*lemma*) and the part of speech (*PoS*) of each word along with the word itself, each component separated with a pipe (|) symbol.

```
word_token | lemma | PoS
```

Byte-pair encoding is implemented after factorization. After training the model, the test dataset on the source side of the language pair is used to obtain the output dataset on the target side of the language pair. Once testing is done, the data is again decoded using the trained BPE model before.

For the **supervised** task, we submit a German to Upper Sorbian factorized model on the German side of the parallel corpus which resulted in 40.9 and 40.3 cased BLEU score.

For the **unsupervised** task, Upper Sorbian to German factorization on the best-performing SS-NMT model improves the BLEU score by 0.1 to 9.0 on Test20 in comparison to the non-factorized model.

The results of the factored models are reported in Tables 3 (supervised) and 4 (unsupervised).

BPE	de (fac.)→hsb	hsb (fac.)→de
2k	31.01	37.09
5k	**41.15**	32.17
10k	35.67	**38.23**
20k	34.70	37.62

Table 3: Supervised Source Factored NMT systems with BLEU scores on DevTest20.

System	BLEU
de→hsb (fac.)	5.67
de (fac.)→hsb (fac.)	6.03
hsb→de (fac.)	7.24
hsb (fac.)→de (fac.)	**7.49**

Table 4: Unsupervised Factored NMT systems with BLEU scores for 10k BPE on DevTest20.

4.3 Data Mining with LASER

We use LASER (Schwenk, 2018; Artetxe and Schwenk, 2019) to filter and mine parallel sentences from a list of monolingual corpora of both German and Upper Sorbian. For German, we use the Wiki_{de}, EP_mono_{de}, and NC_{de} corpora, while

[8] https://github.com/explosion/spaCy

for the Upper Sorbian counterpart, we use the monolingual corpora Ins_{hsb}, Witaj_{hsb}, Web_{hsb}, and Wikipedia dumps (Wiki_{hsb}) as mentioned in Table 1. We explore a range of LASER extraction threshold values (1.03, 1.04, 1.05, 1.06, and 1.07) for this process. Table 5 gives a summary of the number of parallel sentences extracted from the monolingual corpora combinations from both languages using different threshold values. Using a lower threshold value extracts a higher number of parallel sentences but the quality gradually deteriorates as the threshold value decreases. We train NMT models on parallel sentences from each threshold and find that 1.04 gives comparatively better results than others. We use the model M_4 from iterative backtranslation (Table 2) as the baseline and then add the extracted sentences to check if the performance improves. However, all the resulting BLEU scores using additional LASER data are much lower than those of the iterative backtranslation baseline models reported in Table 2, indicating poor quality of the LASER extractions.

Threshold	# Sentences
1.03	18,979
1.04	**9,609**
1.05	5,200
1.06	2,806
1.07	1,646

Table 5: Number of parallel sentences mined using LASER with different threshold values.

4.4 Pre-training with Cross-lingual Language Model XLM

We explore the option of using pre-trained models with different embedding sizes to improve the performance of our system in the unsupervised task. We collected the sentence pairs from Wikipedia extracted with SSNMT. Also we collected back-translations for the monolingual data provided for the task using iterative backtranslation as explained in Section 4.1. We then pre-train XLM for de-hsb using all the monolingual data except Wiki_{de} and Wiki_{hsb}. We then fine-tune the pre-trained model for the supervised translation task using the parallel data from M_0 and back-translations taken from M_4 and M_5. Table 6 shows the resulting BLEU scores for this task on Dev20 and DevTest20.

XLM Embedding Size	BLEU	
	hsb→de	*de→hsb*
Dev20		
512	8.84	**8.41**
1024	**8.91**	8.15
DevTest20		
512	**7.58**	**7.29**
1024	7.44	6.78

Table 6: BLEU scores of pre-training with XLM on Dev20 and DevTest20.

5 Results

Tables 7 (submitted systems) and 8 (unfactored baseline systems) show a summary of all BLEU scores.

Model	BLEU		
	Dev20	DevTest20	Test20
Unsupervised			
de→hsb	13.6	9.9	10.3
hsb→de	11.3	8.1	8.9
hsb (fac.)→de	9.8	8.7	9.0
Supervised			
de (fac.)→hsb	44.34	41.15	40.9

Table 7: BLEU scores for the submitted models on the Dev20, DevTest20, and Test20 datasets.

Unsupervised Parallel data extracted with self-supervised NMT on Wikipedia dumps data and iterative back-translation on mono_{hsb} EP_mono_{de} were used to train the models. For the unsupervised track, we submit three NMT models trained in the directions from unfactored German to unfactored Upper Sorbian (*de→hsb*), from unfactored Upper Sorbian to unfactored German (*hsb→de*), and from factored Upper Sorbian to unfactored German (*hsb*(fac.)→*de*). The iterative backtranslation model M_5 (Table 2) for *de→hsb* obtains a BLEU score of **9.0** on the WMT blind test data. The *hsb→de* model (M_4 in Table 2) achieves a BLEU score of **8.9** while the same model with a factored Upper Sorbian source slightly pushes the BLEU score to **9.0**.

Supervised For the supervised task, we submit a single *de*(fac.)→*hsb* NMT model (refer Table 3) where the German side is factored. The model achieves a BLEU score of **40.9** on the WMT blind test data.

6 Discussion

System	BPE	de→hsb	hsb→de
PBSMT		36.93	37.65
bilingual de-hsb	2k	**41.16**	**40.57**
	5k	37.51	37.47
	15k	37.68	36.79
	30k	36.02	35.64
multilingual de-cs-hsb	2k	28.20	30.98
	5k	34.05	36.07
	15k	32.98	36.61
	30k	29.31	36.89

Table 8: Supervised NMT systems with BLEU scores on DevTest20.

We experimented with different methods in this shared task for both the supervised as well as unsupervised tracks. The major challenge in this task was the small amount of good quality training data as Upper Sorbian is a very low resource language. Parallel sentence extraction demands the availability of good quality data. Schwenk (2018) and Artetxe and Schwenk (2019) mention that the pre-trained LASER model seems to generalize well for minor languages and dialects including Sorbian[9], but Upper Sorbian itself is not among the languages on which the model is actually trained. As a result, LASER does not seem to give very good results for Upper Sorbian. SSNMT (Ruiter et al., 2019) however was able to learn better semantic representations and extracted quality sentence pairs from Wikipedia articles.

The lack of sufficient data for training is also one of the reasons why pre-trained language models using XLM did not give satisfactory results. The second reason is the low quality of the back-translations that were used for fine-tuning.

We have used factored machine translation where we include the *lemma* and the *PoS* of each word along with it in the corpora. Due to the lack of a proper lemmatizer for Upper Sorbian, we used UDPipe (Straka et al., 2016) for Czech as it is another language from the Slavic family. However, there are obvious linguistic differences in both the languages due to which a Czech morphological tool will not work perfectly for Upper Sorbian. This is also the reason why our *de-cs-hsb* multilingual NMT systems (Table 8) did not achieve satisfactory results. NMT models with factored source sentences improved the performances of our models by a small margin.

We have observed that a smaller BPE vocabulary is generally better for low-resource languages as expected. Here we have chosen an optimal BPE vocabulary size as choosing even smaller BPE size values would result in almost character-level segmentation. We also realise that the availability of more quality data could have improved our systems as we can first pre-train language models on good quality monolingual text data using XLM and use this as the initial model for iterative backtranslation as in the SSNMT approach. We believe that this will generate better results.

7 Conclusion and Future Work

This paper describes the UdS-DFKI submission to the shared task of unsupervised and very low resource supervised machine translation between German and Upper Sorbian at WMT20. For all our systems, we have used the standard Transformer-base architecture. We have extracted parallel data from Wikipedia dumps using SSNMT (Ruiter et al., 2019), followed by iterative back-translation for the unsupervised task. For the supervised track, we have tried to factor morphological information into our data to improve our results further. For the constrained supervised task, we achieve 40.9 BLEU for *de*(fac.)→*hsb*. We obtain BLEU scores of 10.3, 8.9, and 9.0 for the *de*→*hsb*, *hsb*→*de*, and *hsb*(fac.)→*de* translation directions respectively in the unsupervised track.

As discussed in Section 6, one approach for future work is to combine XLM pre-training along with SSNMT directly to improve system initialization. It would be interesting to explore linguistic and syntactic information from other closely-related languages to further enhance the performance of the multilingual models.

Acknowledgments

The authors thank the German Research Center for Artificial Intelligence (DFKI GmbH) for pro-

[9] https://github.com/facebookresearch/LASER#supported-languages

viding the necessary infrastructure to run all the experiments.

References

Željko Agić and Ivan Vulić. 2019. Jw300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203.

Saptarashmi Bandyopadhyay. 2019. Factored neural machine translation at loresmt 2019. In *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, pages 68–71.

Saptarashmi Bandyopadhyay. 2020. Factored neural machine translation on low resource languages in the covid-19 crisis.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 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.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation. *CoRR*, abs/1609.04621.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali–english and sinhala–english. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6100–6113.

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.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

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. *CoRR*, abs/1706.03872.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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.

Martin Porter. 2001. Snowball: A language for stemming algorithms.

Dana Ruiter, Cristina Espana-Bonet, and Josef van Genabith. 2019. Self-supervised neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1828–1834.

Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–234.

Rico Sennrich and Barry Haddow. 2016a. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91.

Rico Sennrich and Barry Haddow. 2016b. Linguistic input features improve neural machine translation. *CoRR*, abs/1606.02892.

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

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.