NRC-CNRC Systems for Upper Sorbian–German and Lower Sorbian–German Machine Translation 2021

Rebecca Knowles[*] and **Samuel Larkin**[*]
National Research Council Canada
{Rebecca.Knowles, Samuel.Larkin}@nrc-cnrc.gc.ca

Abstract

We describe our neural machine translation systems for the 2021 shared task on Unsupervised and Very Low Resource Supervised MT, translating between Upper Sorbian and German (low-resource) and between Lower Sorbian and German (unsupervised). The systems incorporated data filtering, backtranslation, BPE-dropout, ensembling, and transfer learning from high(er)-resource languages. As measured by automatic metrics, our systems showed strong performance, consistently placing first or tied for first across most metrics and translation directions.

1 Introduction

This work describes our machine translation (MT) systems for translating between Upper Sorbian–German and Lower Sorbian–German (all translation directions). We focused primarily on the supervised task of Upper Sorbian–German, and then applied those systems to the task of building simple Lower Sorbian–German systems.[1]

Upper Sorbian and Lower Sorbian are Slavic minority languages spoken in eastern Germany, alongside German. The shared task data was provided to the organizers through collaborations with the Sorbian Institute[2] and the Witaj Language Centre,[3] as described in Fraser (2020), to which we direct interested readers for additional information on the languages and data. Following the 2020 shared task, the Witaj Language Centre released a publicly-available Sorbian–German MT system `sotra` (Witaj Language Centre, 2021) based on Moses (Koehn et al., 2007) and OpenNMT (Klein et al., 2017).[4]

We provide an overview of the data, preprocessing, and model architectures in Sections 2, 3, and 4. We then discuss baselines, systems, experiments in monolingual filtering, and backtranslation (all focused on Upper Sorbian–German) in Sections 5, 6, 7, and 8. In Section 9, we discuss how we applied and finetuned our existing Upper Sorbian MT systems for the task of translating Lower Sorbian. Section 10 discusses additional experiments with negative results. Finally, Sections 11 and 12 summarize the final systems and our conclusions.

2 Data

We used all provided parallel German–Upper Sorbian data and all monolingual Upper Sorbian data (after filtering), along with German–Czech parallel data from Open Subtitles (Lison and Tiedemann, 2016),[5] DGT (Tiedemann, 2012; Steinberger et al., 2012), JW300 (Agić and Vulić, 2019), Europarl v10 (Koehn, 2005), News-Commentary v15, and WMT-News.[6] We also used the monolingual Upper Sorbian Web, Witaj and Sorbian Institute datasets as well as the Lower Sorbian monolingual data (the latter for Lower Sorbian tasks only).[7] We used the provided `devel` sets for development, and the `devel_test` systems for measuring progress and choosing which systems to submit.

3 Preprocessing and Postprocessing

As preprocessing, we first clean all of the available training data (but not development or test data) using `clean-utf8-text.pl` with the `-no-phrase-sep` flag from `PortageTextProcessing`.[8] For parallel training data, we use `clean-corpus-n.perl`

[*] Both authors contributed equally to this work.

[1] We abbreviate language names as follows: *cs* (Czech), *de* (German), *dsb* (Lower Sorbian), and *hsb* (Upper Sorbian).

[2] https://www.serbski-institut.de/en/Institute/

[3] https://www.witaj-sprachzentrum.de/

[4] https://sotra.app

[5] http://www.opensubtitles.com

[6] http://www.statmt.org/wmt20/translation-task.html

[7] http://www.statmt.org/wmt21/unsup_and_very_low_res.html

[8] https://github.com/nrc-cnrc/PortageTextProcessing

Data	Lines	BPE	Voc.	CS-DE Parent	Multi.	Multi. ML-0
train.hsb-de.de	60,000	Y	Y		21 ×	36 ×
train.hsb-de.hsb	60,000	Y × 2	Y		21 ×	36 ×
train2021.hsb-de.de	87,521	Y	Y		21 ×	36 ×
train2021.hsb-de.hsb	87,521	Y × 2	Y		21 ×	36 ×
sorbian_institute_monolingual.hsb	337,730	Y × 2	Y		6 × <BT>	
web_monolingual.hsb	105,484				6 × <BT>	
witaj_monolingual.hsb	219,177	Y × 2	Y		6 × <BT>	
OpenSubtitles.cs-de.{de,cs}	11,073,440			Y +10× BPE-dr		
DGT.cs-de.{de,cs}	3,653,397			Y +10× BPE-dr		
JW300.{de,cs}	1,037,533			Y +10× BPE-dr		
Europarl.cs-de.{de,cs}	558,693	Y	Y	Y +10× BPE-dr	3 × <CS>	3 × <CS>
News-Commentary.cs-de.{de,cs}	180,053	Y	Y	Y +10× BPE-dr	3 × <CS>	3 × <CS>
WMT-News.cs-de.{de,cs}	19,892	Y	Y	Y +10× BPE-dr	3 × <CS>	3 × <CS>
news.2019.de.shuffled.deduped.de	31,650,966					
news-commentary-v15.dedup.de	226,820	Y	Y			
news.2019.de.shuffled.deduped.ml_t00	5,071,268					1 x <BT>

Table 1: Data and how it was used, whether for BPE training and vocabulary extraction, parent model training, or child model training. All numbers of lines reflect data after initial cleaning and filtering by known characters. Special tags (for language or backtranslation) are shown where they are used, upsampling is shown with ×, and BPE-dropout is shown.

from Moses (Koehn et al., 2007) with ratio 15, and for monolingual data we remove empty lines. We normalize punctuation with Moses's `normalize-punctuation.perl` and remove the non-breaking space `\xa0`. We perform additional sentence splitting to improve tokenization,[9] then tokenize with Moses's `tokenizer.perl -a -l $LNG` (where $LNG is cs, de, or hsb), then re-merge the sentences that were split into single lines. For all German-Czech parallel data and all monolingual German or Czech data, we removed any lines that contained characters that had not been observed in DE-HSB training data, WMT-News, or Europarl. This helps clean data of unusual encoding issues, as well as removing text that is clearly in other languages (i.e., written in other scripts).

We build BPE vocabularies of size 10k, 15k, 20k, and 25k merges using `subword-nmt`[10] (Sennrich et al., 2016). We also add all Moses and Sockeye special tags (ampersand, <unk>, etc.) and a number of additional reserved tags (for backtranslation, languages, etc.) to a glossary file used for applying BPE, which prevents them from being segmented. For building the BPE models, we used all HSB-DE data, the Sorbian Institute and Witaj monolingual HSB data, CS-DE data, and news-commentary (DE) data; the HSB data was upscaled twice (see Table 1 for full details). The same datasets were used for extracting the joint vocabulary, which was then used for source and target.

In standard postprocessing, we de-BPE and detokenize (using the Moses `detokenizer.perl -a -l $LNG`).

4 Models

We built Transformer models (Vaswani et al., 2017) using Sockeye (Hieber et al., 2018) version 2.3.14 and cuda-10.1. We used the default value of 6 encoder/decoder layers, 8 attention heads, the Adam (Kingma and Ba, 2015) optimizer, label smoothing of 0.1, a cross-entropy-without-softmax-output loss, and a model size of 512 units with a FFN size of 2048. We performed early stopping after 32 checkpoints without improvement. We chose custom checkpoint intervals of 4000 updates when the train corpus was deemed big enough and 500 updates when the train corpus was small. We optimized for BLEU (Papineni et al., 2002)[11] and used the whole validation set during validation. The batch size was set to 8192 tokens, and the maximum sequence length for both source and target was set to 200 tokens. We used weight tying and vocabulary sharing, but we set gradient clipping to absolute and kept the initial learning rate of 0.0002. We used a beam size of 5 in all submit-

[9] Using `utokenize.pl` with `-p -ss -notok -paraline -lang=en` from `PortageTextProcessing`.

[10] `https://github.com/rsennrich/subword-nmt`

[11] All BLEU scores were computed using `sacreBLEU` (Post, 2018) with the signature `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.14`. The chrF scores (Popović, 2015) were generated in the submission interface.

ted systems.[12] When systems deviate from these (i.e., different learning rates or label smoothing), we make note of it in our descriptions.

5 Baselines

We build several types of baselines against which to measure our improvements. They are shown in Table 2 and discussed in the following sections.

System	DE-HSB	HSB-DE
Translation Memory	16.3	15.5
2020 Bitext Baseline	47.0 (10k)	45.7 (10k)
Bitext Baseline	53.2 (10k)	51.9 (15k)
2020 Final Submission	59.4 (20k)	58.9 (10k)

Table 2: BLEU scores for baseline systems measured on devel_test data (vocabulary size in parentheses).

5.1 Translation Memory

We build translation memory baselines, following Simard and Fujita (2012). For each source sentence in devel_test, we find the most similar (as measured by sentence-level BLEU) source sentence in the full training set and return its translation as our hypothesis. The relatively high scores obtained demonstrate the high levels of similarity between the devel_test and train domains (though we note that very few sentences are *exact* matches for ones in the training data).

5.2 Bitext Baselines

We build baselines using the available DE-HSB bitext, first with only the bitext available for the 2020 iteration of the task, and then with the 2020 and 2021 training data combined. As we see in the middle rows of Figure 2, the increase in data from 60,000 to 147,521 lines resulted BLEU score increases of +6.2 in both translation directions.

5.3 2020 Final Systems

As a last "baseline", we consider our final submissions to the 2020 shared task. In the DE-HSB direction, this was a four system ensemble, all of which were child systems incorporating backtranslation and built on top of parent systems trained on either DE-CS or DE-"pseudo-HSB", with a mix of types of BPE-dropout. The HSB-DE direction was an ensemble of 5 child systems using backtranslation on top of a similar set of parent systems. These are described in detail in Knowles et al. (2020).

6 Systems

Here we describe the general types of systems that we have built, including parent DE-CS systems, multilingual systems, and the child (and grandchild) systems we built on top of those.

6.1 Parent Systems

We first built DE-CS and CS-DE parent systems, using OpenSubtitles, DGT, JW300, Europarl, News-Commentary, and WMT-News as training data. The training data is used once in its original form, and concatenated with 10 different versions each generated by an iteration of BPE-dropout (both source and target)[13] with a dropout rate of 0.1. We use newstest2019-csde as the development set. This results in DE-CS and CS-DE systems with BLEU scores between 22 and 25 on the newstest2019-csde development set. We use these parent systems for transfer learning.

6.2 Multilingual Systems

When we build CS-DE and DE-CS parent systems and then use them for transfer learning by finetuning on HSB-DE or DE-HSB data, they undergo "catastrophic forgetting" (Thompson et al., 2019; Gu and Feng, 2020) and lose the ability to translate Czech while gaining the ability to translate Upper Sorbian, as measured on their respective development sets. While we don't necessarily *need* to maintain the ability to translate Czech, we explored whether multilingual systems might improve performance on our task of interest. To this end, we build multilingual systems, which incorporate CS-DE data, upsampled HSB-DE data, and backtranslated data (DE in the case of HSB-DE systems, HSB in the case of DE-HSB systems). In these systems we performed upsampling with the aim of having approximately 1 part CS-DE to 4 parts HSB-DE data (reflective of our priority to translate HSB-DE). We did not experiment with additional ratios; we leave this to future work.

For DE-HSB multilingual systems, we used monolingual HSB data (backtranslated and upsampled 6 times) tagged with <BT> tags, DE-CS data (Europarl, News-Commentary, WMT-News; upsampled 3 times) tagged with <CS>, and the parallel DE-HSB training data (upsampled 21 times and untagged). For HSB-DE multilingual systems,

[12] In paraphrasing experiments where we generated 10-best lists, we used a beam size of 10, but these did not contribute to our final systems.

[13] Note that we only apply BPE-dropout to training data, never to development or test data.

we used a sample[14] of the news 2019 DE data backtranslated and tagged with <BT>, the same three CS-DE corpora tagged with <CS> and upsampled, and the HSB-DE training data upsampled (the upsampling of these latter corpora depended on the size of the backtranslated data).

6.3 Child Systems

Child systems are initialized with the parameters of some given system and are then finetuned on a new set of data with continued training. This is how we perform transfer learning, taking a parent system trained on CS-DE (or DE-CS) data and converting it to an HSB-DE (or DE-HSB) system by starting from the parent system parameters and training on the appropriate language data, as in Kocmi and Bojar (2018). In some cases, we repeat this process multiple times with different sets of data, building "grandchild" systems on top of child systems.

7 Monolingual Data Filtering

Given the tight coupling between the domain of the development/development-test data and the training data, the large quantity of monolingual data available for backtranslation from German, and inspired by the filtering used in the high-performing 2020 submission by Scherrer et al. (2020) we examined whether we should subsample data for backtranslation.[15] We used the 2020 final systems described in Knowles et al. (2020) to backtranslate all available News 2019 DE. This enabled us to train child systems with a random sample of 1.5 million lines of text, the full available backtranslated data, and several approaches to sampling the data.

We first describe HSB-DE experiments with the fixed data size of 1.5 million lines of backtranslated DE monolingual data. We use a *random* sample as a baseline. We compare to it two approaches to using pretrained Sentence-Transformer embeddings[16] (Reimers and Gurevych, 2019) and cosine similarity for domain filtering: ranking sentences in the monolingual data based on their similarity to the *average* embedding of the full DE side of the DE-HSB training data and selecting the 1.5 million most similar,[17] and selecting the 1.5 million sentences most similar to any *individual* sentence in the DE side of the DE-HSB training data.[18] We also apply *Moore-Lewis* filtering (Moore and Lewis, 2010), again treating the DE side of the DE-HSB training data as the "in-domain" data. Moore-Lewis (M-L) uses language models[19] to compare out-of-domain data to in-domain data on the basis of cross-entropy, enabling the sampling of in-domain-like text from the out-of-domain set.

Filter	Both	Src.	None
Random	57.5	57.1	*57.3*
Average	57.5	*57.4*	57.1
Individual	**57.7**	56.9	57.1
Moore-Lewis	**57.7**	57.2	*57.3*
M-L thresh.: 0	**57.7**	**58.1**	**57.9**

Table 3: BLEU scores of HSB-DE child systems trained on authentic HSB-DE parallel text (upsampled) and 1.5 million lines of backtranslated (iteration 1) News 2019 data, sampled using different approaches. Results shown are for 15k vocabulary. Columns indicate type of BPE-dropout (both source and target, source only, and neither). The last line shows thresholded Moore-Lewis, with 5,071,268 lines selected.

We find that the Moore-Lewis approach outperforms or matches the random baseline across three variations of BPE-dropout. Table 3 shows results for 15k BPE, but we found the same across 10k, 15k, 20k, and 25k vocabularies. With both source and target BPE-dropout, the Moore-Lewis sample was always best or tied for best, with source side or no dropout it was always best or second best.

We also built systems with no BPE dropout, using full backtranslated News 2019 data; for larger BPE sizes, the Moore-Lewis samples outperformed the full data (despite being much smaller and thus more efficient), while for the smaller BPE sizes, Moore-Lewis came in second behind the full data.

With 1.5 million as a relatively arbitrary size, we proceeded with using a threshold for Moore-Lewis filtering. A threshold of 0 resulted in 5,071,268 lines sampled from News 2019. With upsampling

[14] As described in Section 7, Moore-Lewis filter.

[15] Data subsampling or filtering or of one sort or another was also used by several other submissions in 2020, including: Dutta et al. (2020), Edman et al. (2020), and Knowles et al. (2020).

[16] From https://github.com/UKPLab/sentence-transformers, with model *paraphrase-xlm-r-multilingual-v1*, a multilingual version of *paraphrase-distilroberta-base-v1*, trained on parallel data for 50+ languages (Reimers and Gurevych, 2020).

[17] Similar to the domain-cosine approach in Aharoni and Goldberg (2020).

[18] We note that this uses external pretrained models, and we have done this only for the purpose of experimenting with backtranslation; none of our final submissions are built using these approaches, so they remain constrained.

[19] 4-gram language models built with MITLM (https://github.com/mitlm/mitlm).

HSB-DE data to match it in size, we found that the Moore-Lewis threshold child models outperformed the 1.5 million size and also outperformed or matched the full data size systems.

We also tested Moore-Lewis filtering on the Upper Sorbian monolingual data, but found it to be less useful in that case, likely due to the much smaller size of available data, and potentially to closer matches (due to the shared origins of the training data and some of the monolingual data).

8 Backtranslation

8.1 BT1

For our first iteration of backtranslation, BT1, we use our final submitted systems from last year's task, as described in Section 5.3.

8.2 BT2

Our second iteration of backtranslation was performed using ensembles of (at the time) best-performing systems at the midpoint of the shared task. Keeping in mind that ensembles typically outperform single systems, and that we found that diverse ensembles seemed to outperform less diverse ensembles, we chose our best-performing systems and then two variants of each to ensemble for the second round of backtranslation. For DE-HSB, the child systems ensembled were the `both`, `src`, and `none` BPE-dropout variants trained on the true DE-HSB data (upsampled 3 times) and BT1 backtranslated HSB data, with a 25k vocabulary. For HSB-DE, the child systems ensembled were also `both`, `src`, and `none` BPE-dropout variants trained on the the true DE-HSB data (upsampled 35 times) and BT1 News 2019 DE data filtered using Moore-Lewis and a threshold of 0, with a 15k vocabulary.

8.3 Analysis of Backtranslation

We compared BT1 and BT2 outputs and found them to be quite similar, sometimes even identical. This brought us to a closer examination of the backtranslation systems and the training data itself. As part of our analysis and experiments, we performed backtranslation of the full DE-HSB training data.

Doing so, we observed that significant portions of the training data had been memorized by many of our systems, and where differences existed, they tended to be quite small. As evidence of this, for both BT1 and BT2, in both translation directions, the BLEU scores for backtranslated training data were 98.2 or higher. Nevertheless, the high automatic metric scores on held-out data suggest that these systems are still able to generalize (that is, they have not *only* memorized data), though it does raise questions about *how* general the models are: would they perform nearly as well on out-of-domain data?

9 Lower Sorbian

The data provided for Lower Sorbian consists of 145,196 lines of monolingual data and the small (approx. 600 line) parallel devel and devel_test sets. In order to build systems, we relied on the relatedness of Lower Sorbian and Upper Sorbian. Since we primarily focused on Upper Sorbian, our BPE vocabularies were *not* learned using Lower Sorbian; we leave an exploration of that to future work. Here we describe our process of building Lower Sorbian systems from Upper Sorbian systems.

9.1 Initial Round

Without any parallel data, we first tried simply translating with our existing HSB-DE and DE-HSB systems and ensembles. In the DE-DSB direction, the resulting devel_test DSB scores were between 7.7 and 8.1 BLEU, while in the DSB-DE direction, the scores were naturally a bit higher (since the system *has* trained on the output language of German), between 17 and 19 BLEU.

From there, we translated the full DSB monolingual data using one of our best HSB-DE single systems: 25k vocabulary, standard parent CS-DE (BPE-dropout both), finetuned child system using BT2 M-L threshold 0 news data and the original HSB-DE training data with BPE-dropout (both) and label smoothing of 0.15. The relatively high BLEU scores that we observed when translating DSB devel and devel_test data with HSB-DE systems allowed us to assume that the output might be more than just noise, and ideally at least good enough for use as the source side.

For backtranslation of DE into DSB, we used an ensemble of two 25k vocabulary DE-DSB systems. The first started from a default parent DE-CS system with BPE-dropout (both) and was then finetuned as a multilingual system using BT2 backtranslated HSB monolingual data and DE-CS data (as described in Section 6.2) with BPE-dropout (both). Then it was finetuned with the initial round backtranslated DSB monolingual data just described, again with BPE-dropout (both). The second also started from the default DE-CS BPE-dropout (both) system and was finetuned with BT2 backtranslated HSB monolingual data with BPE-dropout (both) and learning rate of 0.0001. This was then also finetuned with initial round backtranslated DSB monolingual data, with BPE-dropout (both) and a learning rate of 0.0001.

9.2 Next Round

We also built another DSB-DE system for performing next round backtranslation. It began with a 20k vocabulary standard parent CS-DE (BPE-dropout both), as with previous systems, finetuned child system using BT2 M-L threshold 0 news data and the original HSB-DE training data with BPE-dropout (both) and label smoothing of 0.15. We then finetuned this with the DE side of the full HSB-DE training data, backtranslated to DSB using the initial round DE-DSB system.

10 Inconclusive and Negative Results

We now discuss negative results, i.e., experiments that we performed that were unsuccessful. All of these were performed on Upper Sorbian-German.

10.1 Fuzzy Matching

We performed brief and ultimately unsuccessful experiments with using similar translations from the training data to guide translation, as in Xu et al. (2020). In this approach, for each source sentence (train, development, or test), we first extract its best "fuzzy match" from a translation memory (we use the parallel HSB-DE data for this, and select the best *non-exact* fuzzy match) if any is available. The system input then consists of the source sentence, followed by a special token, followed by the target language sentence corresponding to the closest source fuzzy match from the translation memory (called **FM**# in Xu et al. (2020)). We also tried an approach like their **FM*** approach, where target language tokens are masked with a special token if they do not align[20] to a source language word that is contained in the source sentence to be translated. In either case, if no fuzzy match is returned, a special null token replaces the target language text in the input. Both approaches performed almost identically to the baseline, so we did not proceed with additional experiments (including those approaches that used factors). We experimented with a range of thresholds for fuzzy matches (0.0, 0.35, 0.5), all using `FuzzyMatch-cli`[21] but all performed comparably. We believe this remains an open area for exploration: did the systems fail to outperform the baseline because the baseline had already attained high quality? Did the small size of the translation memory hurt performance?

10.2 Backtranslation as Paraphrasers

Inspired by work like Khayrallah et al. (2020), we also experimented with whether we could treat our high-quality backtranslation systems as paraphrasers to generate more diverse data by translating the HSB-DE parallel data (in each direction) with sampling rather than using one-best output of beam search. We tried building children with this data (both with only authentic target side data and with full combinations of sampled datasets), but did not find that it improved over comparable systems. One issue is that the HSB-DE training data is nearly memorized, as discussed in Section 8.3, so even in the sampled data, many of the differences between translations are quite small.

10.3 Backtranslation-Only Systems

Following Abdulmumin et al. (2021), we experimented with finetuning our parent systems using *only* backtranslated data, followed by then finetuning on the authentic parallel data. We had mixed results with this approach – one of them was high-performing enough to include in our HSB-DE final ensemble, but there was not enough evidence for this language pair to conclude that the approach is broadly useful (beyond providing additional diversity to ensembles).

11 Final Systems

According to preliminary automatic metric results from the shared task organizers, our systems performed quite well. The metrics considered were BLEU, chrF, and – in the case of translation into German – BERT Score. Each translation direction saw five systems submitted, with the exception of DSB-DE, which only had four. Our HSB-DE had the best BERT score (0.981), the second-best BLEU score (67.3, 0.4 BLEU behind NoahNMT), and the best chrF score; it was significantly better than all four other systems in terms of BERT score, while in terms of BLEU and chrF it was better than three other systems (tying with NoahNMT).

[20] We used `fast_align` (Dyer et al., 2013).

[21] `https://github.com/SYSTRAN/fuzzy-match`

	Test		devel_test	
System	**BLEU**	**chrF**	**BLEU**	**BLEU (sing.)**
HSB-DE	67.3 (-0.4)	0.836 (+0.002)	60.0	58.5
DE-HSB	66.3 (+0.4)	0.837 (+0.004)	59.9	58.1
DSB-DE	33.5 (+0.2)	0.638 (+0.016)	34.9	34.5
DE-DSB	29.9 (+2.4)	0.599 (+0.020)	31.0	30.1

Table 4: Final submission scores on test sets. In parentheses, we show the difference between our system and the best performing system by another task participant (positive indicates our system scored highest, negative indicates that the other team's score was higher). The last two columns show scores on devel_test, first for the ensemble and then for the single best component systems in the ensemble (sing.).

Our DSB-DE system had the best BLEU and chrF scores and was tied for the best BERT score (0.953) with CL_RUG; in terms of BLEU and chrF it was significantly better than one other system (alongside CL_RUG and LMU) and in terms of BERT score it was significantly better than two other systems (alongside CL_RUG). Both of our systems translating out of German had the highest BLEU and chrF scores. Our DE-HSB system was, alongside NoahNMT, significantly better than three other systems in both automatic metrics. Our DE-DSB system was, alongside LMU, significantly better than three other systems in both automatic metrics.

11.1 HSB-DE

Our Upper Sorbian-German submission is an ensemble of eight systems with 25k vocabulary, which scored 67.3 BLEU (0.836 chrF) on the test set. The first six systems in the ensemble are children and grandchildren of a CS-DE system (with both source and target BPE-dropout). The final two were multilingual systems trained on a mix of CS-DE and HSB-DE data. Their details are as follows:

1. HSB-DE data, BT2 news (M-L threshold 0), BPE-dropout (both), and label smoothing set to 0.15 (best single system)
2. HSB-DE data, BT2 news (M-L threshold 0), BPE-dropout (both), and transformer dropout at 0.20
3. Child of system 1, finetuned on only HSB-DE (with BPE-dropout, both)
4. Multilingual (mix of CS-DE language-tagged, BT1 news M-L top 1.5M tagged as BT, and HSB-DE upscaled)
5. HSB-DE data, BT1 and BT2 (M-L threshold 0, each), BPE-dropout (both)
6. Child of a backtranslation-only, BT1 and BT2 (M-L threshold 0, each), BPE-dropout (both) system; finetuned on only HSB-DE (with BPE-dropout, both)
7. Multilingual (not a child) mix of CS-DE and HSB-DE data, BT2 (M-L threshold 0), BPE-dropout (both)
8. Multilingual (not a child) mix of CS-DE and HSB-DE data, BT1 (M-L top 1.5M), BPE-dropout (both)

11.2 DE-HSB

Our German-Upper Sorbian system is an ensemble of seven systems with 25k vocabulary, of which the first five are children or grandchildren of a DE-CS parent system (with both source and target dropout). The final two are multilingual systems. In all cases, any backtranslation listed (BT1 or BT2) is backtranslation of the monolingual HSB data. For this language direction, our primary submission does *not* use additional postprocessing. While the additional postprocessing improved all other language directions/pairs, it decreased BLEU by 0.1 in this pair (chrF remained unchanged). The system scored 66.3 BLEU (0.837 chrF) on test.

1. Multilingual system with BT2 and BPE-dropout (both)
2. Child of system 1, finetuned on DE-HSB with BPE-dropout (both)
3. DE-HSB data, BT2, BPE-dropout (both), label smoothing set to 0.15
4. Same as system 3, with transformer dropout set to 0.15
5. DE-HSB data, BT1, BPE-dropout, both
6. Multilingual (not child), BT2, BPE-dr. (both)
7. Multilingual (not child), BT1, BPE-dr. (both)

11.3 DSB-DE

Our Lower Sorbian-German system is an ensemble of two systems with 20k vocabulary, scoring 33.5 BLEU (0.6388 chrF) on test. Both systems are children of the 20k vocabulary equivalent of the first component of the HSB-DE ensemble: taking a CS-DE parent, then training on HSB-DE data and BT2 news (M-L threshold 0), with BPE-dropout (both), and label smoothing set to 0.15.

They train on authentic DE data that is paired with DSB backtranslations, generated by the initial round DE-DSB ensemble (described in Section 9.1).

1. Child trained on backtranslation paired with DE side of the DE-HSB training data, BPE-dropout (both), learning rate 0.0001.
2. Same as the first system, with the addition of backtranslated news 2019 (M-L threshold 0).

11.4 DE-DSB

Our German-Lower Sorbian system is an ensemble of four 20k BPE systems, and scored 29.9 BLEU (0.599 chrF) on test. All four systems are based on a default DE-CS parent with BPE-dropout (both). The first three are then finetuned with BT2 backtranslated HSB data and BPE-dropout (both). The last was finetuned with a multilingual system (again with BT2 backtranslated HSB and BPE-dropout both). We now describe how those systems were finetuned to the DE-DSB task (all used BPE-dropout both):

1. Finetuned with DSB monolingual data (initial round backtranslated).
2. Finetuned with DSB monolingual data (next round backtranslated).
3. Same as system 2 with learning rate 0.0001.
4. Same as system 3 (different parent).

12 Conclusions

As with last year's task, we found that our best systems consisted of ensembles, with more diverse ensembles performing better than less diverse ones. The very high automatic metric scores along with our experiments in backtranslation led us to examine the state of memorization of the training data, which we found to be quite high. We also found that the close relationship between Upper Sorbian and Lower Sorbian enabled us to bootstrap seemingly strong Lower Sorbian systems through iterative backtranslation. We believe that the true test of these systems will be through human evaluation, as well as an analysis of how well they perform in a real-life setting (i.e., with more out-of-domain test data), as the current set seems potentially quite constrained in domain.

Acknowledgements

We thank the anonymous reviewers for their comments and suggestions. We also thank our colleagues Michel Simard, Cyril Goutte, Marc Tessier, Darlene Stewart, Chi-kiu Lo, and Patrick Littell for discussion, comments, feedback, and technical assistance.

References

Idris Abdulmumin, Bashir Shehu Galadanci, and Aliyu Garba. 2021. Tag-less back-translation.

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, Florence, Italy. Association for Computational Linguistics.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Sourav Dutta, Jesujoba Alabi, Saptarashmi Bandyopadhyay, Dana Ruiter, and Josef van Genabith. 2020. UdS-DFKI@WMT20: Unsupervised MT and very low resource supervised MT for German-Upper Sorbian. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1092–1098, Online. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Lukas Edman, Antonio Toral, and Gertjan van Noord. 2020. Data selection for unsupervised translation of German–Upper Sorbian. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1099–1103, Online. Association for Computational Linguistics.

Alexander Fraser. 2020. Findings of the WMT 2020 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 765–771, Online. Association for Computational Linguistics.

Shuhao Gu and Yang Feng. 2020. Investigating catastrophic forgetting during continual training for neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4315–4326, Barcelona, Spain (Online).

International Committee on Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. The sockeye neural machine translation toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 200–207, Boston, MA. Association for Machine Translation in the Americas.

Huda Khayrallah, Brian Thompson, Matt Post, and Philipp Koehn. 2020. Simulated multiple reference training improves low-resource machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 82–89, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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

Rebecca Knowles, Samuel Larkin, Darlene Stewart, and Patrick Littell. 2020. NRC systems for low resource German-Upper Sorbian machine translation 2020: Transfer learning with lexical modifications. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1112–1122, Online. 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, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *Proceedings of the 10th Machine Translation Summit (MT Summit)*, pages 79–86.

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.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

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. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yves Scherrer, Stig-Arne Grönroos, and Sami Virpioja. 2020. The University of Helsinki and aalto university submissions to the WMT 2020 news and low-resource translation tasks. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1129–1138, Online. Association for Computational Linguistics.

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

Michel Simard and Atsushi Fujita. 2012. A poor man's translation memory using machine translation evaluation metrics. In *Proceedings of the 10th Biennial*

Conference of the Association for Machine Translation in the Americas. Association for Machine Translation in the Americas.

Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. 2012. DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 454–459, Istanbul, Turkey. European Language Resources Association (ELRA).

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth 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.

Witaj Language Centre. 2021. Sorbisches Übersetzungsprogramm „sotra" ist online.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.