Abstract

We present our submissions to the WMT21 shared task in Unsupervised and Very Low-Resource machine translation between German and Upper Sorbian, German and Lower Sorbian, and Russian and Chuvash. Our low-resource systems (German↔Upper Sorbian, Russian↔Chuvash) are pre-trained on high-resource pairs of related languages. We fine-tune those systems using the available authentic parallel data and improve by iterated back-translation. The unsupervised German↔Lower Sorbian system is initialized by the best Upper Sorbian system and improved by iterated back-translation using monolingual data only.

1 Introduction

In this paper, we describe systems for translation between German (de) and Upper Sorbian (hsb), German (de) and Lower Sorbian (dsb), and Russian (ru) and Chuvash (cv) developed at LMU Munich for the WMT21 shared task on unsupervised and very low resource machine translation (MT).

Upper Sorbian is a minority language spoken by around 30,000 people in today’s German federal state of Saxony, Lower Sorbian has around 7,000 speakers and is spoken in the German federal state of Brandenburg. With such a small number of speakers, machine translation and automatic processing of Sorbian language 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 Western Slavic languages related to Czech and Polish, it is possible to take advantage of relatively rich resources collected for these two languages.

Unlike our last year’s submission for Upper Sorbian (Libovický et al., 2020), we decided not to use synthetic data from unsupervised translation between Czech and Upper Sorbian and only did iterative back-translation. Despite having more authentic parallel data than last year, our system reaches approximately the same translation quality. Our Upper Sorbian systems ranked third out of six systems in the official ranking.

We leverage the relatedness between the Sorbian languages and use the Upper Sorbian system as a starting point for iterative back-translation using monolingual data only. Our Lower Sorbian Systems ranked second (de↔dsb) and third (dsb↔de) out of four teams in the official ranking.

Chuvash is a minority language spoken in the Volga region in the southwest of Russia. Although it uses the Cyrillic script, it is not related to eastern Slavic languages, but it is a Turkic language, relatively isolated in the Turkic language family. As a language with the highest number of speakers in this shared task, it also has the highest amount of available parallel data. We adopt a similar approach as for German-Upper Sorbian translation and pre-train our models on the related Kazakh language. In addition, we experiment with character-level models in the hope that they will be particularly effective for agglutinative morphology.

2 Experimental Setup

Most of our experimental setup is shared across all the language pairs. All our models use the Transformer architecture (Vaswani et al., 2017) as implemented in FairSeq (Ott et al., 2019).

All data is segmented using BPE (Sennrich et al., 2016b) with 16k merge operations as implemented in YouTokenToMe\(^1\) without previous explicit tokenization. The merges are computed using a concatenation of all training data: German, Czech, Upper and Lower Sorbian in the first set of experiments, Russian, Kazakh, and Chuvash in the second set of experiments.

For the supervised task, we first pre-train mod-
els on high-resource related languages: Russian-Kazakh for Chuvash and German-Czech for Upper Sorbian. We first train Transformer Base models on authentic data. These systems are used to generate back-translation (Sennrich et al., 2016a) of monolingual data. Using tagged back-translation (Caswell et al., 2019), we trained Transformer Big models for German↔Czech and Russian↔Kazakh translation. All back-translation steps use sampling and simple length-based filtering as proposed by Edunov et al. (2018). We upsample the authentic parallel data to match the size of the synthetic data.

We keep most default hyperparameters from the predefined architectures in FairSeq (transformer for the Base model, transformer_wmt_en_de_big_t2t for the Big model). The batch size is 6k tokens for the Base models, 2k tokens for Big models on a single GPU. Because we always start with high-resource training, we keep the dropout on the standard value of 0.1.

We use these models to initialize the weights (Nguyen and Chiang, 2017; Kocmi and Bojar, 2018) of the supervised low-resource models without restarting the optimizer. Because the learning rate is already low at that stage of training, we do not need to change the dropout to prevent overfitting. First, we train the supervised models using the authentic parallel data only, then we continue with iterated back-translation. The best Upper Sorbian-to-German model is used to translate Lower Sorbian monolingual data into German. In the next steps, we continue with a standard iterative back-translation procedure for unsupervised neural machine translation (Artetxe et al., 2018; Lample et al., 2018).

Our final submission is an ensemble (with the vote strategy) of the best-scoring systems in the process of iterated back-translation. Language-pair specific descriptions and results are discussed in the following sections.

We evaluate our systems using the BLEU Score (Papineni et al., 2002), chrF score (Popović, 2015) as implemented in SacreBLEU (Post, 2018). Further, we evaluate the models using BERTScore (Zhang et al., 2020) with XLM-RoBERTa Large (Conneau et al., 2020) as an underlying model for German and Russian and mBERT (Devlin et al., 2019) for Chuvash. Similar to the official task evaluation, we also report for each system the number of significantly worse systems in each metric at the significance level 0.95 with bootstrap resampling (Koehn, 2004) with 1k samples. For each metric, each system receives one point for each system it significantly outperforms in the metric at the significance level of 0.95.

3 German ↔ Upper Sorbian

Pre-training. For training the German↔Czech systems, we followed the same setup as in our last year’s submission (Libovický et al., 2020). We used all parallel datasets from the Opus project (Tiedemann, 2012), which was 15.4M sentences after filtering by length and language identity. We trained a Transformer Base model on this data and used this model to generate back-translation. We used 20M Czech and 20M German sentences from the WMT News Crawl. We mix the back-translated and authentic parallel data one-to-one and train Transformer Big models on it.

---

Figure 1: A diagram of the training procedure of the German↔Upper/Lower Sorbian systems. Gray dashed arrows (→) denote model initialization, solid black arrows (→) denote synthetic data generation by back-translation.

---

2We re-used the published code https://github.com/pytorch/fairseq/tree/master/examples/backtranslation.

3BLEU score signature nrefs:1|case:mixed| eff:no|tok:13a|smooth:exp|version:2.0.0
chrF score signature nrefs:1|case:mixed|eff:yes| nc:6|nw:0|space:no|version:2.0.0

4https://github.com/Tiiiger/bert_score
**Sorbian data.** We used all Upper Sorbian data provided for the shared task, i.e., 148k parallel sentence pairs (this is 88k sentence pairs more than last year), we did not apply any filtering on the parallel dataset. The development validation and the development test set of 2k sentences were the same as the last year.

**Back-translation.** We used 15M German sentences from the WMT News Crawl and all available monolingual Upper Sorbian data, 696k sentences, for back-translation. We applied the same rule-based statistical fixing of hyphenation-related OCR errors as the last year (Libovický et al., 2020, § 3.1). To better leverage the limited amount of monolingual data, we sample the Upper Sorbian translations $5 \times$. We iterated the back-translation 4 times, always initializing the model with the Czech-German models (see Figure 1).

**Results.** The results are presented in Table 1. In the translation direction into German, the translation quality gradually increased between the back-translation steps. In the opposite direction, the translation quality oscillated. We attribute this to a larger amount of authentic German sentences. Ensembling only has a negligible effect. Note also that for translation into Sorbian, no differences between the models are statistically significant. In the opposite direction, the BLEU and the chrF score only separate the systems into two clusters, whereas the differences among BERTScores are always significant in the bootstrap testing, even though the absolute score differences are smaller. The best system for translation into German is a single from the last iteration of back-translation despite scoring slightly worse in the BLEU score.

### Table 1: Quantitative results of the German↔Upper Sorbian translation systems on the development test data.

<table>
<thead>
<tr>
<th></th>
<th>hsb → de</th>
<th></th>
<th></th>
<th>de → hsb</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>chrF</td>
<td>BERTScore</td>
<td>Points</td>
<td>BLEU</td>
<td>chrF</td>
</tr>
<tr>
<td>Authentic data only</td>
<td>53.4</td>
<td>.763</td>
<td>.933</td>
<td>0</td>
<td>54.9</td>
<td>.769</td>
</tr>
<tr>
<td>BT iter 1</td>
<td>55.2</td>
<td>.773</td>
<td>.936</td>
<td>1</td>
<td>56.4</td>
<td>.778</td>
</tr>
<tr>
<td>BT iter 2</td>
<td>55.8</td>
<td>.777</td>
<td>.937</td>
<td>4</td>
<td>56.5</td>
<td>.778</td>
</tr>
<tr>
<td>BT iter 3</td>
<td>55.8</td>
<td>.777</td>
<td>.937</td>
<td>5</td>
<td>56.2</td>
<td>.778</td>
</tr>
<tr>
<td>BT iter 4</td>
<td>56.1</td>
<td>.779</td>
<td>.938</td>
<td>7</td>
<td>56.0</td>
<td>.776</td>
</tr>
<tr>
<td>Ensemble</td>
<td>56.2</td>
<td>.779</td>
<td>.938</td>
<td>6</td>
<td>56.4</td>
<td>.779</td>
</tr>
</tbody>
</table>

Table 2: Automatic scores for the best German↔Lower Sorbian Systems.

<table>
<thead>
<tr>
<th></th>
<th>dsb→de</th>
<th></th>
<th></th>
<th>de→dsb</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>chrF</td>
<td>BERTScore</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsb→de Single</td>
<td>33.7</td>
<td>.606</td>
<td>.873</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsb→de Ensemble</td>
<td>33.8</td>
<td>.602</td>
<td>.874</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>de→dsb Single</td>
<td>30.1</td>
<td>.587</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>de→dsb Ensemble</td>
<td>30.1</td>
<td>.588</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: chrF scores during iterative back-translation for unsupervised German↔Lower Sorbian translation. The orange vertical lines denote 95%-confidence intervals using bootstrap resampling.

**German ↔ Lower Sorbian**

**Data.** Because this is a purely unsupervised task, we did not use any Lower Sorbian parallel data.

4 German ↔ Lower Sorbian

We used the same German monolingual data as we used for back-translation for Upper Sorbian. We use all the Lower Sorbian monolingual data, 145k sentences, provided by the organizers.

**Iterative back-translation.** Similarly to Upper Sorbian, we sample the back-translation of Lower Sorbian $10 \times$ for higher diversity in the training data.

**Results.** The final results are tabulated in Table 2. Figure 2 shows the translation quality in terms of chrF score during back-translation iterations. Similar to Upper Sorbian, the direction into German that uses larger monolingual data tends to improve more smoothly than the opposite direction. Also, the ensembling of the three best-scoring systems only has a negligible effect. The single system and
the ensemble do not significantly differ in any of the metrics.

5 Russian ↔ Chuvash

Pre-training. Similar to Upper Sorbian systems, we pre-train the systems on high-resource related language pair, Kazakh-Russian. We used the crawled Kazakh-Russian corpus of 5M sentence pairs published for WMT19 (Barrault et al., 2019) to train a Transformer Base model. We used these models to back-translation 3M Kazakh and 3M Russian sentences from the WMT News Crawl from the most recent years.

Chuvash data. We used all parallel data provided by the organizers, 717k sentence pairs, without any filtering. For back-translation, we used all 2.8M monolingual Chuvash sentences provided for the competition. For Russian, we used 18M monolingual sentences from the WMT News Crawl.

Back-translation. We ran two iterations of back-translation. We sample from the model during back-translation. We sampled 4 different translations for each Chuvash sentence to increase the training data diversity. We mix the authentic and synthetic parallel training data in the one-to-one ratio. All models are initialized by the Russian↔Kazakh models.

Character models. We further experiment with finetuning the system to the character level. Libovický and Fraser (2020) managed to train a character-level system for another Turkic language, English-to-Turkish translation. Here, we test if this is a property of Turkic languages or an artifact of the dataset English-Turkish dataset. We follow Libovický and Fraser (2020) and finetune the subword model to the character level.

Table 3: Quantitative results of the Russian↔Chuvash translation systems on the development test data.

Results. The results are presented in Table 3. Compared to other language pairs, back-translation had a surprisingly small effect on the translation quality. We suspect this result might be due to errors in data processing or signalize a need for a better data filtering technique. Model ensembling has no effect here. The character-level systems are on average 2 BLEU points worse than their subword counterparts, which is consistent with the results of character-level models on high-resource languages (Libovický and Fraser, 2020). Surprisingly, the character-level models seem to have much larger gains from model ensembling than the subword-based models. In fact, the ensemble of the character-level models is statistically indistinguishable from the best subword-based models.

6 Conclusions

We presented our systems for low-resourced translation between German and Upper Sorbian, unsu-
supervised translation between German and Lower Sorbian, and translation between Chuvash and Russian.

Our systems used standard state-of-the-art techniques for low-resource and unsupervised machine translation but did not exhaust all available methods. Better results could be achieved using more monolingual data and by more careful filtering of the synthetic parallel data.

Acknowledgments

This work was also supported by the European Research Council under the European Union’s Horizon 2020 research and innovation program (grant agreement #640550) and by the DFG (grant FR 2829/4-1).

References


A Training hyper-parameters

We use the following command line options for fairseq-train command in all experiments. For the Transformer Base models, we use the pre-defined transformer architecture, for Transformer Big, we use transformer_wmt_en_de_big_t2t. The batch size is 6000 tokens for the Base models and 2000 tokens for the Big models.

```bash
code snippet
fairseq-train \
   $DATA \ 
   --arch $ARCHITECTURE \ 
   --share-all-embeddings \ 
   --label-smoothing 0.1 \ 
   --criterion \ 
      label_smoothed_cross_entropy \ 
   --optimizer adam \ 
   --adam-betas '(0.9, 0.998)' \ 
   --clip-norm 5.0 \ 
   --lr 5e-4 \ 
   --lr-scheduler inverse_sqrt \ 
   --warmup-updates 16000 \ 
   --max-tokens $TOKENS
```

References


