A Simplified Training Pipeline for Low-Resource and Unsupervised Machine Translation

Àlex R. Atrio[1,2] and **Alexis Allemann**[1] and
Ljiljana Dolamic[3] and **Andrei Popescu-Belis**[1,2]

[1]HEIG-VD / HES-SO
Yverdon-les-Bains
Switzerland
name.surname@heig-vd.ch

[2]EPFL
Lausanne
Switzerland

[3]Armasuisse, W+T
Thun
Switzerland
ljiljana.dolamic@armasuisse.ch

Abstract

Training neural MT systems for low-resource language pairs or in unsupervised settings (i.e. with no parallel data) often involves a large number of auxiliary systems. These may include parent systems trained on higher-resource pairs and used for initializing the parameters of child systems, multilingual systems for neighboring languages, and several stages of systems trained on pseudo-parallel data obtained through back-translation. We propose here a simplified pipeline, which we compare to the best submissions to the WMT 2021 Shared Task on Unsupervised MT and Very Low Resource Supervised MT. Our pipeline only needs two parents, two children, one round of back-translation for low-resource directions and two for unsupervised ones and obtains better or similar scores when compared to more complex alternatives.

1 Introduction

Several known techniques enable the design of neural MT systems with little or no parallel data for the source and target languages. Among them are the initialization with a parent model trained on parallel data from related languages (Zoph et al., 2016; Kocmi and Bojar, 2018) and repeated cycles of back-translation of monolingual data that create pseudo-parallel corpora used for training (Sennrich et al., 2016a; Hoang et al., 2018). When designing a very low-resource or unsupervised system, many practitioners rightfully consider as a guideline the best-performing systems found in several shared tasks, such as WMT Shared Task on Unsupervised MT and Very Low Resource Supervised MT (Fraser, 2020; Libovický and Fraser, 2021a; Weller-Di Marco and Fraser, 2022), where teams compete in order to obtain the highest scores among them. While these systems typically do obtain very high scores, in this paper we show that the pipelines of the highest-scoring systems in this task may be unnecessarily complex, and they can be substantially simplified while still achieving comparable results.

To solve this shared task, high-resource parent models have been leveraged to initialize child models for low-resource languages, which in turn have been used to warm-start the training for unsupervised directions. However, the submissions to the above-mentioned shared task typically developed several dozen models, with numerous parent/child models in both directions as well as increasingly better models trained on several rounds of back-translated data. These models were finally ensembled for best results.

For the 2021 edition of the task, the unsupervised language pair was Lower Sorbian / German (DSB/DE), with parallel data only available for testing, while the low-resource pair was Upper Sorbian / German (HSB/DE). A large amount of German / Czech (DE/CS) parallel or monolingual data is available to train parent models, due to the similarity of Sorbian dialects to Czech. Moreover, given the similarity of the two Sorbian dialects, child low-resource models can become parents of "grandchild" systems for the unsupervised task. As a result, these systems are quite complex, which raises the question: up to which point can these architectures be simplified with virtually no loss of performance?

Our study answers this question by presenting a simpler pipeline than the ones submitted to the shared task, which reaches superior or comparable scores to the ones from the highest-scoring teams. In our pipeline, we apply the same selection and filtering of data as the best-performing team for comparability. We train high-resource parent models on authentic parallel data in two directions (CS↔DE), and then use them to initialize child low-resource models (HSB↔DE). We improve these systems with one round of back-translated monolingual data, and finally use them to initialize systems and to produce back-translated

Proceedings of the The Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023), pages 47–58

data for the unsupervised pair (DSB↔DE). More specifically, our simplifications are the following:

1. only training from one initialization per parent-child-grandchild;
2. no multitasking and no multilingual models;
3. length-based filtering of back-translated data instead of language model-based one;
4. no monolingual data and only moderate amount of authentic parallel data for high-resource parent models;
5. a single round of back-translation for low-resource directions and two for unsupervised directions;
6. same subword vocabulary for all translation directions;
7. moderately-sized Transformer-Base instead of Big;
8. unique set of values for hyper-parameters such as learning rate and label smoothing.

We make public the configuration files that create these systems in the OpenNMT-py framework.[1]

2 Related Work

2.1 Techniques for Low-Resource and Unsupervised MT

Transfer learning consists in training a model on a high-resource pair (parent) that initializes a model trained on a lower-resource one (child). Initially, Zoph et al. (2016) kept the same target language between parent and child. Kocmi and Bojar (2018), however, showed that the identity or relatedness of the target languages is not essential, and that all of the weights of the child systems can be initialized with those of the parent model without changing the training routine.

Back-translation consists in automatically translating monolingual data in the target language, in order to create a synthetic parallel corpus which can be used for training (Sennrich et al., 2016a). Edunov et al. (2018) showed that the benefits of back-translated data depend on the decoding algorithms used to generate it, and that beam search is not the best-performing option unless the amount of data to back-translate is small. This, however, can be mitigated by differentiating authentic and synthetic data with tags (Caswell et al., 2019). This process can also be performed iteratively, as shown by Hoang et al. (2018), with either the same model generating initial back-translated data, improving its performance, and re-generating the data, or by training a new model for each round of back-translation, which improves the quality of the synthetic data.

When large monolingual corpora are available, fully unsupervised NMT can be achieved by using masked language modeling, denoising, or translation language modeling (Lample et al., 2017, 2018; Conneau and Lample, 2019). This results in cross-lingual language models (Conneau and Lample, 2019), which can further be trained on back-translated data. Such systems perform best when jointly trained on very large monolingual datasets and when a small amount of parallel data is available (Song et al., 2019; Liu et al., 2020). However, this is not the case for some of the datasets of the WMT shared task considered here.

2.2 Submissions to the WMT21 Shared Task

Six teams competed for the highest scores in the low-resource Upper Sorbian / German and the unsupervised Lower Sorbian / German translation tasks at the WMT 2021 Shared Tasks on Unsupervised MT and Very Low Resource Supervised MT (Libovický and Fraser, 2021a). The datasets used in the tasks are presented in Section 4.1 below. The organizers scored the submissions using automatic metrics over held-out test sets. NRC-CNRC (Knowles and Larkin, 2021) and LMU (Libovický and Fraser, 2021b) achieved some of the highest scores in both tasks. Other competitive scores were achieved by CL_RUG (Edman et al., 2021) and NoahNMT (Zhang et al., 2021), followed at some distance by IICT-Yverdon (Atrio et al., 2021). Since no team participated in both tasks, and NoahNMT used a particularly complex pipeline with very large amounts of training data and a pre-trained BERT encoder, we decided to work towards the simplification of the NRC-CNRC and LMU 2021 pipelines.

The NRC-CNRC submission (Knowles and Larkin, 2021) experimented with various numbers of BPE merges (Sennrich et al., 2016b) for different translation directions and for generating synthetic data for training. Their final vocabularies contain 25k and 20k subwords for the supervised and unsupervised models, respectively. They built the BPE tokenizer from upscaled HSB, CS and DE data, but without DSB. The architecture is

[1] github.com/AlexRAtrio/simplified-pipeline

based on Transformer-Base (Vaswani et al., 2017), with frequent ensembling throughout the pipeline. They use Moore-Lewis filtering (Moore and Lewis, 2010) of back-translated sentences. They train parent CS↔DE models on the entire parallel CS-DE data in Table 1, with BPE-dropout (Provilkov et al., 2020). From them, they initialize child HSB↔DE models, which are further fine-tuned into grandchildren DSB↔DE.

The final HSB→DE system from NRC-CNRC is an ensemble of eight different models. Six of them are children and grandchildren of CS-DE models, and two are multilingual CS-DE and HSB-DE models (with no transfer learning). Among the other six, there are different values for hyper-parameters like learning rate or label smoothing. After training with various filtering strategies for back-translated sentences, Moore-Lewis filtering was found to perform best, although differences are generally smaller than 1 BLEU point. Some models are fine-tuned only with back-translations, or only authentic data, or both. For DE→HSB translation, the translation is generated with an ensemble of seven systems. The final NRC-CNRC submission to the DSB→DE unsupervised task is an ensemble of two grandchild systems trained with different back-translated corpora, and for DE→DSB it is an ensemble of four grandchildren, with different rounds of back-translation, different learning rates, and at least one different CS-DE parent model.

The LMU submission (Libovický and Fraser, 2021b) starts with a BPE tokenizer with 16k merges, on the entire HSB, DE, CS and DSB data. Parent Transformer-Base CS↔DE models are trained on the entire CS-DE parallel data, which is filtered by length and language identity. To this authentic data, they add 20M lines of monolingual CS and DE respectively for back-translation, which they use to train another set of parent models with Transformer-Big, sampling and tagged back-translation. Child HSB→DE and DE→HSB models (also Transformer-Big) are trained from CS-DE parents, first on authentic parallel data. Then, they are used to iteratively back-translate 15M lines of DE and the entire HSB monolingual data for four rounds, with a new model initialization for each round. To obtain DSB→DE and DE→DSB grandchildren systems, iterative back-translation is performed for eight rounds, initialized from the respective HSB/DE Transformer-Big child systems.

A similar shared task was again organized at WMT 2022, including HSB↔DE and DSB↔DE translation (Weller-Di Marco and Fraser, 2022). Additional parallel HSB-DE data was provided, increasing the total to about 0.5 million lines, which likely increased scores for the low-resource supervised tasks HSB↔DE. Moreover, an unsupervised HSB↔DE and a low-resource supervised DSB↔HSB translation tasks were introduced.

Four teams participated in the low-resource supervised tasks, and three in the unsupervised ones. In most tasks, HuaweiTSC (Li et al., 2022) achieved by far the highest scores, thanks to a deep 35-layer encoder, 6-layer decoder Transformer (Wei et al., 2021) and a parent multilingual model trained on vast amounts of data (including 55M lines of DE-CS, 66M lines of DE-PL, and 20M of monolingual DE). In addition to the techniques we study in this paper, Li et al. (2022) used regularized dropout (Liang et al., 2021) to improve consistency while training. Their setup thus also consisted of numerous and expensive training steps, just as the NRC-CNRC and LMU systems to which we compare our proposal.

3 Proposed Pipeline

We propose a simplified training pipeline represented in Figure 1, which reaches comparable or better results than the above systems. The pipeline is minimal, in the sense that only eight systems are trained for HSB↔DE and DSB↔DE translation, including parent systems for initialization. We show that one round of back-translation for low-resource directions and two for unsupervised ones are sufficient. In comparison with the numerous rounds and checkpoints of the NRC-CNRC and LMU systems, our pipeline is an order of magnitude smaller.

We start by training from scratch parent models DE→$\text{CS}_{\text{parent}}$ and CS→$\text{DE}_{\text{parent}}$ on authentic parallel data. From their best-performing checkpoint, we respectively initialize DE→$\text{HSB}_{\text{authentic}}$ and HSB→$\text{DE}_{\text{authentic}}$ models, which we train only on authentic parallel data. We then use their best-performing checkpoints to generate synthetic parallel data (back-translations) by translating monolingual target data (resulting in synthetic datasets HSB_{BT}-DE_{mono} and DE_{BT}-HSB_{mono}). We initialize from the best-performing checkpoints of the previous systems new models DE→$\text{HSB}_{\text{authentic+BT}}$ and HSB→$\text{DE}_{\text{authentic+BT}}$ which we train on upscaled authentic parallel data and back-translated

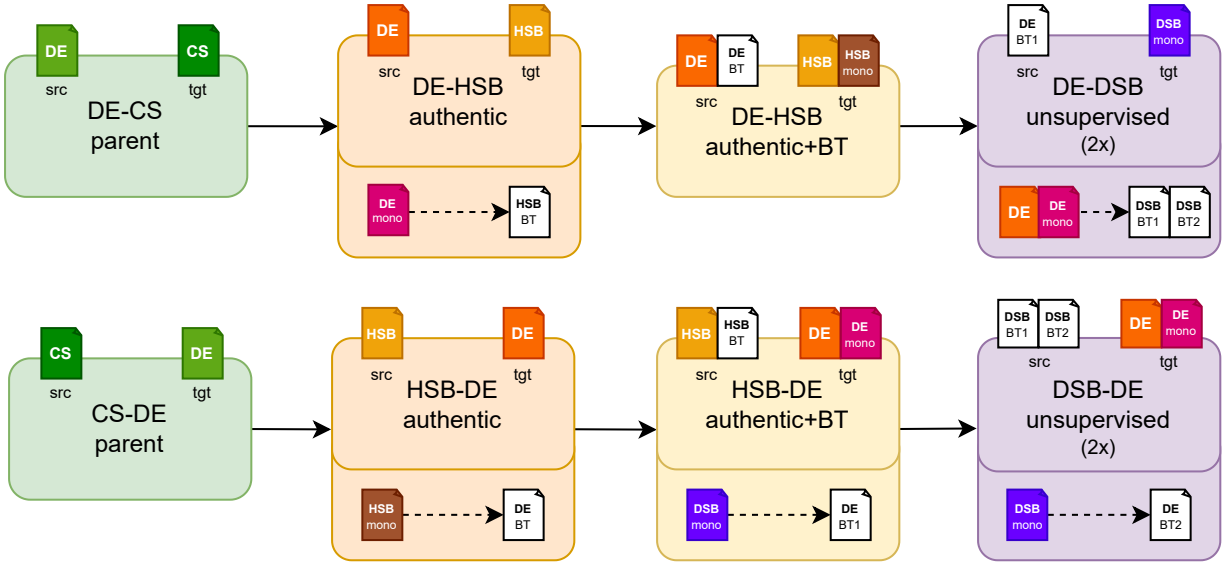

Figure 1: Pipeline of implemented systems. Solid arrows represent the parent systems used, and dashed arrows represent creation of synthetic data through back-translation. The datasets in color are those presented in Table 1. The datasets in white, to the right of dashed lines, are the back-translations (BT) generated by our systems. The unsupervised models are trained with two rounds of back-translation.

data.

Finally, with the best-performing checkpoint of system HSB→$DE_{authentic+BT}$, we perform back-translation of monolingual DSB data (resulting in DE_{BT1}-DSB_{mono}), and train with this first round of synthetic parallel data the unsupervised DE→$DSB_{unsupervised(a)}$ model. We use this system for the first round of back-translation in the opposite direction, of the DE part of the HSB-DE authentic data and monolingual DE (resulting in DSB_{BT1}-DE and DSB_{BT2}-DE_{mono}) into DSB, on which we train the unsupervised DSB→$DE_{unsupervised(a)}$ model. We then use this system for the second round of back-translation of monolingual DSB data and train another unsupervised DE→$DSB_{unsupervised(b)}$ model, and with it we perform a second round of back-translation of monolingual DE to train a final unsupervised DSB→$DSB_{unsupervised(b)}$ model.

ID - Language	Dataset name	Size (sentences)
DE-CS	DGT v8	4,894
	Europarl v8	569
	JW300	1,039
	News Comm. v16	197
	OpenSubtitles	16,358
	WMT-News	20
DE-HSB	WMT 2020 Train	60
	WMT 2021 Train	88
HSB_{mono}	WMT20 Sorbian Inst.	340
	WMT20 Web	133
	WMT20 Witaj	222
DSB_{mono}	WMT21 Mono.	145
DE_{mono}	WMT21 News Crawl 19	1,500

Table 1: Monolingual and parallel corpora with their languages as presented in Figure 1. We provide the number of lines (sentences) after filtering, in thousands.

4 Data, Preprocessing and Systems

4.1 Corpora

The datasets we use are listed in Table 1, and the identifiers correspond to those in Figure 1. They encode the language and index number for authentic parallel DE-CS, authentic parallel DE-HSB, and monolingual HSB, DSB, and DE. For the CS↔DE parent models we use parallel data from DGT (Tiedemann, 2012; Steinberger et al., 2012), Europarl (Koehn, 2005), JW300 (Agić and Vulić, 2019), OpenSubtitles (Lison and Tiedemann, 2016), News Commentary, and WMT-News.[2] Our HSB↔DE models use datasets from the 2020 edition of the task, with monolingual HSB data from three sources: (a) the Sorbian Institute provided a mix of high- and medium-quality HSB data; (b) the Witaj Sprachzentrum provided high-quality HSB

[2] statmt.org/wmt20/translation-task.html

data; (c) the Web data consists of web-scraped noisier HSB data gathered by the Center for Information and Language Processing at LMU Munich (Fraser, 2020). Our DSB↔DE models use only the monolingual Lower Sorbian (DSB) dataset from the 2021 shared task.

To evaluate our systems, we use the 'Newstest2019-csde' as a test set for our CS↔DE models. For our HSB↔DE and DSB↔DE models we use the 'devel' set from the WMT20 task during development, and 'devel_test' for final evaluations. Since the official scores of the task are calculated on an undisclosed subset of the blind test set, we cannot compare our results with the final official ones. We will thus compare them with the scores on 'devel_test' reported by each team in their articles. Our two evaluation metrics are the same as in the shared task. We use the SacreBLEU library (Post, 2018) to compute BLEU (Papineni et al., 2002).[3] We also use BERTScore[4](Zhang et al., 2019), with the XLM-RoBERTa-Large model (Conneau et al., 2020) for translations into German, as provided with the BERTScore toolkit. We test the statistical significance of differences in scores at the 95% confidence level using paired bootstrap resampling from SacreBLEU.

4.2 Data Filtering

For comparison purposes, we follow closely the data preparation procedure of the NRC-CNRC team (Knowles and Larkin, 2021). We first clean the training data with the `clean_utf8.py` script from `PortageTextProcessing`.[5] Subsequently, parallel training data is filtered with the `clean-corpus-n.perl` script from Moses (Koehn et al., 2007) to remove sentence pairs with a length ratio larger than 15. Punctuation is then normalized using the `normalize-punctuation.perl` script from Moses. Finally, non-breaking spaces (Unicode U+00A0 or '`\xa0`') and empty lines are deleted.

For all DE-CS parallel data and all monolingual DE and CS data, lines that contain characters which have not been observed in DE-HSB training data, WMT-News, or Europarl corpora are deleted. This is done to eliminate encoding issues and text that is clearly in other languages. The DE monolingual dataset consists of a likewise cleaned random sample of the full WMT21 News Crawl 19 corpus. The numbers of lines after filtering are shown in the two rightmost columns of Table 1.

4.3 Tokenization

We start tokenizing sentences into words with the Moses tokenizer: `tokenizer.perl -a -l $LNG`, where `$LNG` is `cs` or `de`, using the `cs` code also for HSB and DSB data. Then, we use Byte Pair Encoding (BPE) (Sennrich et al., 2016b)[6] to build a vocabulary of 20k subwords. For building the BPE models, we used all HSB-DE data, the Sorbian Institute and Witaj monolingual HSB data (but not the Web-crawled HSB data, which is too noisy), both sides of CS-DE data, and News-Commentary (DE) data. The HSB data was upscaled twice. The same datasets were used for extracting the joint vocabulary, which was then used to tokenize the source and target sides with a BPE-Dropout rate of 0.1 (Provilkov et al., 2020).

In post-processing, we detokenize BPE subwords with the BPE toolkit and then with a script from Moses: `detokenizer.perl -a -l $LNG`, where `$LNG` is `cs` or `de`, using the `cs` code also for HSB and DSB data.

4.4 System Architecture

We use Transformer models (Vaswani et al., 2017) from the OpenNMT-py library (Klein et al., 2017) version 2.3.0.[7] We use the following default values of hyper-parameters from Transformer-Base: 6 encoder/decoder layers, 8 attention heads, Adam optimizer (Kingma and Ba, 2014), label smoothing of 0.1, dropout of 0.1, hidden layer of 512 units, and FFN of 2,048 units. We share the vocabulary and use the same embedding matrix for both input and output languages. The batch size is 8,192 tokens, and the maximum sequence length for both source and target is 501 tokens. We keep OpenNMT-py's scaling factor of 2 over the learning rate. We use standard values for hyper-parameters in order to maintain a simplified pipeline, although it is likely that a more regularized system could further improve scores (Atrio and Popescu-Belis, 2022).

We do not use any early stopping measure and train for a sufficiently large amount of steps to ensure convergence. We train the parent CS↔DE

[3] github.com/mjpost/sacrebleu, signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1`.

[4] github.com/Tiiiger/bert_score, signature: `xlm-roberta-large_L17_no-idf_version=0.3.12(hug_trans=4.26.0)_fast-tokenizer`

[5] github.com/nrc-cnrc/PortageTextProcessing

[6] github.com/rsennrich/subword-nmt

[7] github.com/OpenNMT/OpenNMT-py

models for 500,000 steps, and the children and grand-children ones for 100,000 steps. To train our models we use between one and four Nvidia RTX 2080 Ti with 11 GB RAM which amounts to around 80 hours for parent models, 30 hours for children models (systems 3/4 and 5/6), and 15 hours for grandchildren models. As better parent systems lead to better children, we trained the parents for a longer time, given also the larger parallel data available.

We save checkpoints every 4,000 steps during training, and obtain the testing scores from an ensemble of the four best checkpoints in terms of BLEU scores on the validation data. When testing, we use a beam size of 5 for all systems, except when indicated otherwise for back-translation.

5 Results of the Proposed Pipeline

5.1 Parent DE↔CS Systems

We first train the DE→$\text{CS}_{\text{parent}}$ and CS→$\text{DE}_{\text{parent}}$ models (see Figure 1) on the authentic parallel CS-DE data presented in Table 1. The BLEU and BERTScore of these systems, shown in Table 2, are respectively 20.2 and 22.1. These are comparable with the ones reported by NRC-CNRC (22–25 BLEU points) and with those with the same architecture appearing in the Opus-MT leaderboard[8], trained on OPUS parallel data (Tiedemann, 2012) using Opus-MT-Train (Tiedemann and Thottingal, 2020).

Choosing Czech for the parent model is reasonable due to its similarity with Upper and Lower Sorbian, but we have found that this similarity is not crucial (Atrio et al., 2021). Using a similar setup, we observed almost identical results with a Polish↔German parent model, and a loss of only 1.3 BLEU points with a French↔German one.

5.2 Child DE↔HSB Systems

We initialize the child systems DE→$\text{HSB}_{\text{authentic}}$ and HSB→$\text{DE}_{\text{authentic}}$ models from the highest-scoring checkpoint of the respective parent, and trained them on authentic parallel HSB-DE data. The systems reached BLEU scores of 56.7 and 56.1 respectively (see Table 2).

One round of back-translation. We hypothesize that due to the already existing authentic parallel data, one round of back-translation (BT) could be sufficient. We use the above systems

[8] opus.nlpl.eu/leaderboard/DE→CS and CS→DE

System	BLEU	BERTScore
DE→CS$_{\text{parent}}$	20.2	.936
CS→DE$_{\text{parent}}$	22.1	.938
DE→HSB$_{\text{authentic}}$	56.7	-
HSB→DE$_{\text{authentic}}$	56.1	.975

Table 2: BLEU and BERTScore on `newstest2019` for CS-DE parent models and `devel_test` for HSB-DE models trained only on authentic data.

to generate synthetic parallel data from monolingual DE and HSB corpora. To generate it, we decode by sampling from the entire model distribution rather than applying beam search, following Edunov et al. (2018). As shown in Figure 1, with the HSB→$\text{DE}_{\text{authentic}}$ and DE→$\text{HSB}_{\text{authentic}}$ systems we translate the DE_{mono} data into HSB_{BT}. Similarly, we translate the HSB_{mono} data into DE_{BT}. Therefore, we obtain two pseudo-parallel datasets with authentic target sides. We apply to them the same filtering process as in Section 4.2, except for a more restrictive cut-off for `clean-corpus-n.perl`, using a maximum ratio of 1.5 between sentences instead of 15. This filtering results in the deletion of respectively 5% and 11% of the HSB-DE and DE-HSB pseudo-parallel datasets.

We continue training the HSB→$\text{DE}_{\text{authentic}}$ and DE→$\text{HSB}_{\text{authentic}}$ systems with authentic parallel HSB-DE data and the back-translated data, with the former being upscaled to match the number of lines of the latter. We obtain respectively the systems noted HSB→$\text{DE}_{\text{authentic+BT}}$ and DE→$\text{HSB}_{\text{authentic+BT}}$. The improvements brought by this round of back-translation are only of about 1 BLEU point (see Table 5). Our scores are similar to those reported by NRC-CNRC without inter-model ensembling (57-58 BLEU). With the highest-scoring checkpoint for each of HSB→$\text{DE}_{\text{authentic+BT}}$ and DE→$\text{HSB}_{\text{authentic+BT}}$ we generate synthetic data for the unsupervised case by translating monolingual DSB and DE.

Iterative back-translation. We found that our pipeline does not benefit from multiple rounds of back-translation thanks to an additional experiment, not included in the final pipeline. Following Libovický and Fraser (2021b), for each round of back-translation i (with $i = a, b, c$), systems HSB→$\text{DE}_{\text{authentic+BT}(i)}$ and DE→$\text{HSB}_{\text{authentic+BT}(i)}$ are respectively initialized from the parent models CS→$\text{DE}_{\text{parent}}$ and DE→$\text{CS}_{\text{parent}}$ trained on

CS-DE data, instead of child systems trained on only authentic data HSB→$DE_{authentic}$ and DE→$HSB_{authentic}$ as performed above. Decoding and filtering remain as described above as well. Otherwise, the first round of back-translation remains as above, and the second round results in new pseudo-parallel datasets on which we train new systems in both directions (also including upscaled authentic parallel data HSB-DE), resulting in systems HSB→$DE_{authentic+BT(b)}$ and DE→$HSB_{authentic+BT(b)}$. We perform a third round to obtain systems HSB→$DE_{authentic+BT(c)}$ and DE→$HSB_{authentic+BT(c)}$. Hence, this method differs from our main proposed pipeline in the usage of three rounds versus one, and the initialization of models from CS-DE parents instead of the child HSB-DE systems trained on authentic parallel data.

While several studies have suggested that multiple back-translation rounds are beneficial, our findings are more nuanced. As we observe in Table 3, for the direction DE→HSB, the first round of back-translation improves BLEU by 1.2 points, but afterwards scores decrease. For the direction HSB→DE, on the contrary, BLEU scores continue to improve with more iterations, although with diminishing returns, with a final improvement of 0.7 points. We hypothesize that this is due to the monolingual DE dataset being larger than the HSB one.

Direction	System	BLEU
DE→HSB	DE→$HSB_{authentic}$	56.7
	DE→$HSB_{authentic+BT(a)}$	**57.9**⋆
	DE→$HSB_{authentic+BT(b)}$	**57.6**⋆
	DE→$HSB_{authentic+BT(c)}$	57.4
HSB→DE	HSB→$DE_{authentic}$	56.1*
	HSB→$DE_{authentic+BT(a)}$	56.5*
	HSB→$DE_{authentic+BT(b)}$	56.5*
	HSB→$DE_{authentic+BT(c)}$	**56.8**

Table 3: BLEU scores for only authentic parallel data, and three rounds of back-translation: DE→HSB systems are trained with $DE_{BT(i)}$-HSB_{mono} and HSB→DE systems are trained with $HSB_{BT(i)}$-DE_{mono}. We note in bold the highest score in each direction. We denote scores that are *not* significantly different per direction with the same symbol.

In contrast, Libovický and Fraser (2021b) observed more significant improvements over four rounds of iterative back-translation, although also with diminishing returns. For HSB→DE, their improvement was 2.7 (up to 56.1 BLEU), starting however from a lower score than ours (53.4) and getting half of the improvement in the first iteration. For the DE→HSB, they achieve a smaller improvement of 1.6, up to 56.5 overall, starting from 54.9. Their highest scores are obtained after two rounds. We hypothesize that the difference between our results and theirs regarding the HSB→DE direction is explained by their use of ten times more monolingual DE data, coupled with a larger architecture.

Following Edunov et al. (2018) we experimented with various decoding methods for the back-translation stage. As a comparison to the full unrestricted sampling we use in all systems, we studied restricted sampling of the top 10 candidates, as well as the dropout of 10% of the words after standard decoding, and their combination. For DE→$HSB_{authentic+BT(a)}$ the three methods obtained nearly identical scores (57.54, 57.54, and 57.51), and none of them substantially deviated from our original method. This supports previous observations by Edunov et al. (2018) showing that differences between decoding algorithms for back-translation are only noticeable when the monolingual data size is large (e.g. more than 8M lines).

5.3 Grandchild DE↔DSB Systems

In contrast with the DE↔HSB low-resource case, we hypothesize that more than one round of back-translation may be useful in the unsupervised case. We used system HSB→$DE_{authentic+BT}$ to create the pseudo-parallel dataset DE_{BT}-DSB_{mono}, with which we trained system DE→$DSB_{unsupervised(a)}$. With this system, we generated synthetic DSB data from the DE part of the HSB-DE authentic data as well as monolingual DE, resulting in DSB_{BT1}-DE and DSB_{BT2}-DE_{mono}. For rounds b and c we repeated the process as with HSB-DE, initializing system DE→$DSB_{unsupervised(b)}$, (and then c) and system DSB→$DE_{unsupervised(b)}$ (and then c), respectively from the highest-scoring checkpoint from systems DE→$HSB_{authentic+BT}$ and HSB→$DE_{authentic+BT}$, and generating synthetic data with each other. Filtering removed between 6-9% of the lines. The scores of the resulting systems are shown in Table 4.

For DE→DSB, the second round of back-translation produced a large improvement of 3.3 BLEU points over the first round, but the third round resulted in a minimal improvement of 0.1. The large improvement of system DE→$DSB_{unsupervised(b)}$ may be explained by the

Direction	System	BLEU
DE→DSB	DE→$\text{DSB}_{\text{unsupervised}(a)}$	26.1
	DE→$\text{DSB}_{\text{unsupervised}(b)}$	**29.4**⋆
	DE→$\text{DSB}_{\text{unsupervised}(c)}$	**29.5**⋆
DSB→DE	DSB→$\text{DE}_{\text{unsupervised}(a)}$	36.5
	DSB→$\text{DE}_{\text{unsupervised}(b)}$	**38.1***
	DSB→$\text{DE}_{\text{unsupervised}(c)}$	**38.4***

Table 4: BLEU scores for three rounds of back-translation: DE→DSB systems are trained with $\text{DE}_{\text{BT}(i)}$-DSB_{mono} and DSB→DE systems are trained with $\text{DSB}_{\text{BT}(i)}$-$\text{DE}_{\text{mono}}$ and $\text{DSB}_{\text{BT}(i)}$-DE (the DE part of the authentic HSB-DE data). The highest score in each direction is in bold. Scores that are *not* significantly different per direction are marked with the same symbol.

fact that the synthetic data used to train it is the first DE set translated by a true DSB system (DSB→$\text{DE}_{\text{unsupervised}(a)}$). For DSB→DE we also observe improvements from several rounds of back-translation, with the second one improving BLEU by 1.6 points and the third round improving only minimally by 0.3 points. We hypothesize that this difference is due to the lower amount of DSB monolingual data versus DE, and the back-translation of the DSB data being generated by a model that had not been trained on DSB. For both directions (DE→DSB and DSB→DE) the difference between systems a and b was significant, but not between b and c. As a result, we excluded extra rounds of back-translation for low-resource HSB-DE from our simplified pipeline, and only performed two rounds for unsupervised DSB-DE.

6 Discussion and Conclusion

We show in Table 5 the final results of our pipeline, compared to the highest scores for each direction obtained in the WMT 2021 shared task (Libovický and Fraser, 2021a). Scores from CFILT (Khatri et al., 2021) are not shown because we do not have access to their 'devel_test' scores. HSB-DE scores from CL_RUG are intermediate scores for their unsupervised DSB-DE systems.

On both low-resource directions (HSB↔DE) our simpler pipeline obtains comparable results to the three highest-scoring teams (NRC-CNRC, LMU and NoahNMT systems). Our scores on one unsupervised direction (DSB→DE) surpass those of the three participants, while on the other (DE→DSB) our scores are comparable to those of the two highest-scoring teams (NRC-CNRC and LMU). To explain the latter result, we hypothesize that our simplified pipeline is more sensitive to weight initialization, and therefore is less robust across all directions than a more complex pipeline.

Compared to the NRC-CNRC submission, our pipeline uses the same data selection and filtering, a single vocabulary for the tokenizer, trains from a single random initialization for each of the translation direction, does not train multitask or multilingual models, uses a much simpler filtering for back-translated sentence pairs, and sets a single set of values for hyper-parameters such as learning rate and label smoothing.

Compared to LMU, our pipeline uses a smaller amount of authentic parallel data for the parent CS↔DE models, does not use monolingual data back-translated for these parent models, and uses an architecture with fewer parameters (Transformer-Base instead of Big). Moreover, we use only one round of back-translation instead of four for the child HSB↔DE systems and two instead of eight for the grandchild DSB↔DE systems submitted by LMU.

NoahNMT also produced high scores on the supervised tasks, although with the use of a pre-trained BERT model (Devlin et al., 2019), vast amounts of monolingual data (100M lines), and dual parent transfer. CL_RUG scored well in the unsupervised tasks, but made use of sequence masking, denoising auto-encoding, cross-lingual back-translation, and vocabulary alignment between HSB and DSB with VecMap (Artetxe et al., 2018). IICT-Yverdon applied a scheduled multi-task training to both the supervised and unsupervised directions, which appeared to be particularly ineffective for the unsupervised task.

We now provide some hypotheses on why our simplified pipeline produces scores that are comparable with those from more complex ones. Firstly, a much better trained parent model does not necessarily result in noticeable better child models. Whatever the cause of the improvement of the parent models (additional parent training data, parent back-translation, or additional parent pairs), when several stages in the training pipeline can be found afterwards (such as training on authentic data, then children back-translation, then grandchildren back-translation, etc.), the initial benefit may be lost later in the pipeline. This is particularly exacerbated when child systems are later trained with data of dubious quality, such as back-translations. Artetxe et al. (2020), for instance, showed that when per-

System	DE→HSB	HSB→DE		DE→DSB	DSB→DE	
	BLEU	BLEU	BERTScore	BLEU	BLEU	BERTScore
NRC-CNRC	**59.9**	**60.0**	-	**31.0**	34.9	-
LMU	56.5	56.2	.938	30.1	33.8	.874
NoahNMT	58.3	58.5	-	-	-	-
CL_RUG	52.1	51.6	-	24.9	32.1	-
IICT-Yverdon	54.6	53.2	-	9.62	-	-
Ours	57.4	57.0	.976	29.4	**38.1**	.958

Table 5: BLEU and BERTScore on the 'devel_test' set of the best-performing system of each team, with our proposals at the bottom. The highest score per direction is in bold. The systems are referenced in Section 2.2 above, and '-' indicates that the score is not available.

forming iterative back-translation, the quality of the initial system has minimal effect on the final performance, as systems tend to converge to scores dictated by the monolingual data.

This first hypothesis feeds into a second hypothesis: large amounts of parent parallel or monolingual data make it reasonable for practitioners to choose larger architectures, which must then be carried over to the lower-resource children, since pruning rarely happens mid-pipeline. Although there is evidence that fitting large models to very small amounts of data is not necessarily detrimental (Belkin et al., 2019) and can even be beneficial (Li et al., 2020), it is unclear if this still holds with a more complex training pipeline. In any case, a smaller architecture in a low-resource setting, while still over-parameterized, can perform as well as a larger one.

As a third hypothesis, and on a more practical note, since it is necessary to carry out the full pipeline to obtain the final results, some practitioners may choose to introduce elements into the pipeline without empirically measuring the extent to which they improve the scores, since that sometimes may require re-training the entire pipeline.

Finally, modern Transformer-based systems are robust, and there seems to be a large area of "acceptable results" which is relatively easy to access, as we have empirically shown with our comparison to five different submissions to the WMT shared task. However, our pipeline is only trained on a group of similar languages (Czech, Upper Sorbian, and Lower Sorbian) to and from German, which may not generalize in the same manner to other languages or domains.

To sum up, although the competition to achieve first place in shared tasks such as the one discussed here leads participants towards increasingly complex pipelines, we have shown that competitive or even better results can be achieved with a much simpler training pipeline. We hope this will encourage practitioners to further participate in shared tasks such as these, while minimizing entry constraints regarding time, training strategy, or computing resources.

Limitations

The simplified pipeline put forward in this paper has demonstrated its merits in one specific context, but should also be tested with different data sizes and differences in language similarity. Although we compared with the main techniques used by the participants, it is possible that other techniques for unsupervised translation based on vector space alignment are also competitive, though this is less likely here given the scarcity of monolingual data for Sorbian.

Ethics Statement

This study does not process personal or sensitive data. While MT in general may facilitate disclosure or cross-referencing of personal information, which may pose threats to minorities, the community appears to consider that the potential benefits far outweigh the risks, judging from the large number of studies for low-resource and unsupervised MT.

Acknowledgments

We thank Armasuisse (UNISUB projet: Unsupervised NMT with Innovative Multilingual Subword Models) and the Swiss National Science Foundation (DOMAT project: On-demand Knowledge for Document-level Machine Translation, n. 175693). We are grateful to the three anonymous LoResMT reviewers for their helpful suggestions.

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, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. 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, Noe Casas, and Eneko Agirre. 2020. Do all roads lead to Rome? understanding the role of initialization in iterative back-translation. *Knowledge-Based Systems*, 206:106401.

Àlex R. Atrio, Gabriel Luthier, Axel Fahy, Giorgos Vernikos, Andrei Popescu-Belis, and Ljiljana Dolamic. 2021. The IICT-yverdon system for the WMT 2021 unsupervised MT and very low resource supervised MT task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 973–981, Online. Association for Computational Linguistics.

Àlex R. Atrio and Andrei Popescu-Belis. 2022. On the interaction of regularization factors in low-resource neural machine translation. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 111–120, Ghent, Belgium. European Association for Machine Translation.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lukas Edman, Ahmet Üstün, Antonio Toral, and Gertjan van Noord. 2021. Unsupervised translation of German–Lower Sorbian: Exploring training and novel transfer methods on a low-resource language. In *Proceedings of the Sixth Conference on Machine Translation*, pages 982–988, Online. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. 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.

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.

Jyotsana Khatri, Rudra Murthy, and Pushpak Bhattacharyya. 2021. Language model pretraining and transfer learning for very low resource languages. In *Proceedings of the Sixth Conference on Machine Translation*, pages 995–998, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *arXiv: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 and Samuel Larkin. 2021. NRC-CNRC systems for Upper Sorbian-German and Lower Sorbian-German machine translation 2021. In *Proceedings of the Sixth Conference on Machine Translation*, pages 999–1008, 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. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

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.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

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.

Shaojun Li, Yuanchang Luo, Daimeng Wei, Zongyao Li, Hengchao Shang, Xiaoyu Chen, Zhanglin Wu, Jinlong Yang, Zhiqiang Rao, Zhengzhe Yu, Yuhao Xie, Lizhi Lei, Hao Yang, and Ying Qin. 2022. HW-TSC systems for WMT22 very low resource supervised MT task. In *Proceedings of the Seventh Conference on Machine Translation*, pages 1098–1103, Abu Dhabi. Association for Computational Linguistics.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. 2020. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *International Conference on machine learning*, pages 5958–5968. PMLR.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 10890–10905. Curran Associates, Inc.

Jindřich Libovický and Alexander Fraser. 2021a. Findings of the WMT 2021 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2021b. The LMU Munich systems for the WMT21 unsupervised and very low-resource translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 989–994, Online. 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).

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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.

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.

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.

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, Berlin, Germany. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 5926–5936.

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).

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).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, Hao Yang, and Ying Qin. 2021. HW-TSC's participation in the WMT 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 225–231, Online. Association for Computational Linguistics.

Marion Weller-Di Marco and Alexander Fraser. 2022. Findings of the WMT 2022 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Seventh Conference on Machine Translation*, pages 801–805, Abu Dhabi. Association for Computational Linguistics.

Meng Zhang, Minghao Wu, Pengfei Li, Liangyou Li, and Qun Liu. 2021. NoahNMT at WMT 2021: Dual transfer for very low resource supervised machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1009–1013, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: evaluating text generation with bert. In *Proceedings of the International Conference on Learning Representations (ICLR 2020)*. arXiv:1904.09675.

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.