The IICT-Yverdon System for the WMT 2021 Unsupervised MT and Very Low Resource Supervised MT Task

Àlex R. Atrio[1,2] and **Gabriel Luthier**[1] and **Axel Fahy**[1] and
Giorgos Vernikos[1,2] and **Andrei Popescu-Belis**[1,2] and **Ljiljana Dolamic**[3]

[1]HEIG-VD / HES-SO
Yverdon-les-Bains
Switzerland

[2]EPFL
Lausanne
Switzerland

[3]Armasuisse, W+T
Thun
Switzerland

name.surname@heig-vd.ch

ljiljana.dolamic@armasuisse.ch

Abstract

In this paper, we present the systems submitted by our team from the Institute of ICT (HEIG-VD / HES-SO) to the Unsupervised MT and Very Low Resource Supervised MT task. We first study the improvements brought to a baseline system by techniques such as back-translation and initialization from a parent model. We find that both techniques are beneficial and suffice to reach performance that compares with more sophisticated systems from the 2020 task. We then present the application of this system to the 2021 task for low-resource supervised Upper Sorbian (HSB) to German translation, in both directions. Finally, we present a contrastive system for HSB-DE in both directions, and for unsupervised German to Lower Sorbian (DSB) translation, which uses multi-task training with various training schedules to improve over the baseline.

1 Introduction

In this paper, we present the systems submitted to the WMT 2021 task on Unsupervised MT and Very Low Resource Supervised MT. We first build a series of baseline systems, driven mostly by considerations of simplicity, trained on data from the 2020 edition of the task, for translation between Upper Sorbian (HSB) and German (DE). These systems, described in Section 3, enable us to quantify the merits of using additional back-translated data (Sennrich et al., 2016) and of initializing the system for a low-resource pair with parameters learned on a high-resource pair (same target language and related source language).

The systems described above serve as the basis for our 2021 baseline submitted to the shared task, for DE→HSB and HSB→DE, presented in Section 4, which improves upon our 2020 baseline with the addition of more parallel data, and achieves competitive performance with the use of back-translation and parent-initialization only. However, this approach does not lead to an effective baseline for unsupervised German to Lower Sorbian (DSB) translation (Section 5). In Section 6, we present experiments with a contrastive system that implements multi-task learning, with several schedules, in which denoising tasks together with translation are presented to the systems in increasing order of complexity, leading to more robust HSB↔DE systems, together with a strategy of diverse ensembling. We also use our DE→HSB system to initialize a multi-task DE→DSB system for the unsupervised task, although in this case the performance is not competitive.

2 Datasets

We use various Upper Sorbian datasets from the 2020 edition of the task, and additional WMT data, as presented in Table 1. The monolingual HSB data from 2020 comes from three sources: `sorbian_institute_monolingual` consists of a mix of high- and medium-quality HSB data provided by the Sorbian Institute; `witaj_monolingual` consists of high-quality HSB data from the Witaj Sprachzentrum; finally, `web_monolingual` consists of web-scraped noisier HSB data gathered by the Center for Information and Language Processing from LMU Munich (Fraser, 2020). We kept from all datasets only sentences that have strictly more than 2 and strictly fewer than 301 words.

3 Baseline HSB→DE System on 2020 Data

3.1 Subword Vocabulary

For the HSB→DE system, we use CS→DE initialization in several experiments, because Czech (CS) is a high-resource language and close neighbor to Upper Sorbian. Therefore, we create a tri-lingual shared subword vocabulary (CS, DE, HSB) using the Unigram LM model (Kudo,

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 973–981
November 10–11, 2021. ©2021 Association for Computational Linguistics

Dataset	**Language**	**Before filtering**		**After filtering**	
		sentences	words	sentences	words
Sorbian Institute Monolingual	HSB	339,822	5,044,079	339,822	5,044,079
Web Monolingual	HSB	121,003	1,661,898	115,632	1,651,154
Witaj Monolingual	HSB	222,027	2,672,255	215,370	2,660,805
Europarl v8	DE	2,234,583	48,430,884	2,186,477	48,347,698
JW300	DE	2,366,722	34,782,112	2,182,801	34,519,064
News Commentary v15	DE	422,009	8,942,517	409,955	8,939,335
Europarl v8 CS-DE	CS	568,589	11,571,876	562,716	11,561,049
Europarl v8 CS-DE	DE	=	13,098,638	=	13,086,320
JW300 CS-DE	CS	1,052,338	13,579,350	982,034	13,435,536
JW300 CS-DE	DE	=	15,133,882	=	14,992,424
News Commentary v13 CS-DE	CS	174,789	3,486,672	172,987	3,479,819
News Commentary v13 CS-DE	DE	=	3,751,102	=	3,746,708
WMT 2020 HSB-DE Train	DE	60,000	724,572	59,030	722,076
WMT 2020 HSB-DE Train	HSB	=	639,740	=	637,883
WMT 2021 HSB-DE Train	DE	87,521	1,251,339	87,502	1,251,287
WMT 2021 HSB-DE Train	HSB	=	1,094,421	=	1,094,375

Table 1: Monolingual and parallel corpora with their languages and numbers of lines (sentences) and words, before and after filtering by length (keeping sentences with more than 2 and fewer than 301 words).

2018) as implemented in SentencePiece.[1] We apply 32,000 merges and the other parameters of SentencePiece are kept to default values. We obtain 600k sentences of HSB data from `sorbian_institute_monolingual`, `witaj_monolingual` and `train.hsb-de`, the latter being the HSB side of the 2020 training data. We do not use `web_monolingual` as it appears to be noisy, due to the collection process. For CS and DE, 600k sentences are selected randomly from the monolingual corpora listed in Table 1. The vocabulary generated by SentencePiece is converted from log probabilities to frequencies using the `spm_to_vocab.py` tool from the OpenNMT-py toolkit. Using a common SentencePiece model for the three languages is not obligatory, but appeared to improve the performance by 2-3 BLEU points in most cases.

3.2 System Parameters and Results

We use OpenNMT-py (Klein et al., 2017) for our experiments.[2] We start with Transformer-Base (Vaswani et al., 2017) (78M parameters) but also experiment with Transformer-Big (245M parameters), with their main parameters described in Table 2. We apply the same regularization and optimization procedures to the two models. We accumulate gradients over 2 batches and train on 2 GPUs, with a `batch_size` of 1k for Base and and 2k for Big. We use the "noam" learning rate schedule (Vaswani et al., 2017) with its values at each step multiplied by two, and 8k warmup steps. We evaluate and save checkpoints every 5k steps. Final translations are generated with a beam width of 5, ensembling the last two checkpoints in these experiments. We report BLEU scores (Papineni et al., 2002) obtained with SacreBLEU (Post, 2018) on detokenized text.

	N	h	d_{model}	d_{ff}	P_{drop}	steps
Base	6	8	512	2048	0.1	60k
Big	6	16	1024	4096	0.3	100k

Table 2: Parameters of the two Transformer models used in our experiments. Other parameters are set to the default values of the OpenNMT-py toolkit.

3.3 Use of Back-translated Data

The first HSB→DE system we trained, for comparison purposes, used only the HSB/DE parallel data provided for the WMT 2020 Low-Resource task. Its BLEU scores are 47.98 on the 'dev' set (`devel.hsb-de`) and 41.22 on the 'devtest' set (`devel_test.hsb-de`) after 60k steps of training (first line of Table 3). The already high BLEU scores that are reached, compared to scores generally observed on high-resource language pairs,

[1] https://github.com/google/sentencepiece (v. 0.1.95)

[2] https://github.com/OpenNMT/OpenNMT-py (v. 2.0.1)

indicate that the ‘dev’ and ‘devtest’ sets are probably quite similar to the training data.

We obtain additional training data through back-translation (Sennrich et al., 2016) of widely available monolingual German data. To this end, we train a DE→HSB model on the same parallel corpus as above, which reaches BLEU scores of 45.23 / 40.62 respectively on ‘dev’ and ‘devtest’. Using this model, we translate News Commentary V15 from German into Upper Sorbian. The resulting pseudo-parallel data (noisy on the HSB side) is used in addition to the initial data for training a new HSB→DE model, which reaches a score of 52.91 / 44.39 (second line of Table 3). The improvement of this single enrichment with imperfect data of the initial low-resource system thus exceeds 4 BLEU points.

3.4 Initialization with Parameters from a High-Resource Pair

The second technique we use for improvement is transfer from a high-resource pair (Zoph et al., 2016; Kocmi and Bojar, 2018), i.e. initialization with parameters from an MT system trained on such a pair. As Upper Sorbian has many similarities with Czech, which is a high-resource language, we initialize the HSB-DE model with the parameters of a model trained for CS→DE, then train it with the same data as in the previous subsection. Firstly, the CS→DE model is trained using Europarl and News Commentary, and reaches a BLEU score of 27.13 on a sample test set extracted from these two corpora.

The resulting HSB→DE system reaches BLEU scores of 55.99 / 47.53, a further increase of about 3 BLEU points (third line of Table 3). The use of an even larger dataset further improves performance: the addition of the JW300 corpus (Agić and Vulić, 2019) to the CS→DE training data increases BLEU by half a point (56.5 on ‘dev’). The rather small increase could be attributed to the large difference in domains between JW300 and the HSB/DE data.

Since back-translation can provide very large amounts of data, we also trained a Transformer Big (with the parameters shown in Table 2) with the addition of the monolingual German corpora of Europarl and JW300 backtranslated into Upper Sorbian. This model reaches 58.08 / 49.99 BLEU points respectively on ‘dev’ and ‘devtest’, improving performance by more than 1.5 BLEU points. This is currently our best baseline model for HSB→DE, obtained with two simple augmentation techniques only.

We can compare this score with three of the highest-scoring systems on the 2020 HSB→DE ‘devtest’ set, noting some of the differences between them and our baseline. Scherrer et al. (2020) achieved a BLEU score of 56.9 using back-translation and bilingual pre-training with CS→DE, but also scheduled multitask with several monolingual and multilingual tasks. Knowles et al. (2020) achieved a BLEU score of 58.9 using iterative back-translation, multiplication of the HSB data for BPE training, and character- and word-level lexical modifications of Czech to make it more similar to Upper Sorbian. Libovický et al. (2020) achieved a score of 56.0 with much larger corpora for back-translation and CS→DE pre-training (14M lines) and the use of an unsupervised CS→HSB system to translate the CS side of the DE/CS parallel data into HSB.

3.5 Initialization with Parameters from Other High-Resource Pairs

We studied the role of the closeness between Upper Sorbian and the high-resource source language used for initialization, by reproducing the above initialization experiments (CS→DE) with Polish and French instead of Czech. Polish is a West Slavic language just as Czech and Upper Sorbian, although geographically more remote, whereas French is a Romance language: we thus expected the former to outperform the latter. To keep training time more manageable, we used a Transformer-Base, and trained the parent model on Europarl and JW300, because News Commentary is not available for Polish. For each experiment we build a different tri-lingual SentencePiece model trained with 600k sentences per language.

The use of the PL→DE model (with a 22.33 BLEU score on its respective test set) for initialization leads to a HSB→DE performance of 56.07 / 47.94, which is very similar to the system initialized with CS→DE parameters (55.99 / 47.53). The use of the FR→DE model (with a 19.25 BLEU score) for initialization leads to a HSB→DE system reaching 54.92 / 46.30. This is about 1.3 BLEU points lower than with Polish or Czech, although the difference is smaller than expected given the linguistic distance between French and Upper Sorbian. These results are in line with the findings of Aji et al. (2020) who argue that no parent is clearly better than other for transfer learning in MT.

System	HSB→DE		DE→HSB	
	dev	devtest	dev	devtest
1. Transformer-Base, 2020 parallel data	47.98	41.22	45.23	40.62
2. Add back-translated data to #1	52.91 *(+4.93)*	44.39 *(+3.17)*	51.00 *(+5.77)*	43.23 *(+2.61)*
3. Initialize #2 with high-resource pair	55.99 *(+3.08)*	47.53 *(+3.14)*	–	–
4. Transformer-Big with #3	58.08 *(+2.09)*	49.99 *(+2.46)*	–	–
5. Add 2021 parallel data to #4	59.29 *(+1.21)*	51.86 *(+1.87)*	57.22 *(+6.22)*	49.95 *(+6.72)*

Table 3: Scores of our 2020 (1–4) and 2021 (5) baseline systems, with absolute improvements brought by each additional technique or data set.

3.6 Two Rounds of Back-Translation

Multiple rounds of back-translation can be done on each side, but this computational effort is not always compensated by a significant increase of the BLEU score. Using the best HSB→DE system above, we translate monolingual HSB data and use it to train an improved DE→HSB model, which reaches 51.00 on the 'dev' data (+5.77 with respect to the initial DE→HSB system) and 43.23 on the 'devtest' data (+2.61). We then use this improved model to translate the monolingual German data again and use the resulting pseudo-parallel data to train a new HSB→DE model. The model without CS initialization reaches BLEU scores of 53.62 on 'dev' (+0.62) and 44.95 on 'devtest' (+0.43). If CS initialization is used, the models reaches respectively 58.44 (+0.36) and 50.03 (+0.04) on 'dev' and 'devtest'. The improvement brought by the additional rounds of back-translation is quite marginal, therefore we do not pursue this approach, and focus on a system which is initialized from a parent high-resource pair and trained with original and back-translated data, where the latter comes from a reverse system trained only with the original parallel HSB-DE data provided by the shared task.

4 Baseline HSB↔DE Low Resource Systems for 2021

Given the results of the previous section, we choose the Transformer-Big for our 2021 baseline. We change the dropout level from 0.3 to 0.1 since our experiments revealed an increase in performance with the latter value. Furthermore, we add the 87,502 sentences of additional parallel HSB-DE training data provided in 2021 to the datasets used in our 2020 baseline. We use the same SentencePiece model with DE, HSB, and CS data that we used for our 2020 baseline system, with approximately 700k lines for each language. At translation time, after observing a number of out-of-vocabulary tokens, we replace the unknown tokens with the source token that has the highest attention weight. We do not make any further changes regarding our 2020 Transformer-Big model.

The scores of our baseline systems on 2020 and 2021 data are synthesized in Table 3 for the various techniques we experimented with. Our baseline HSB→DE model with combined 2021 and 2020 data is system #5 in Table 3: it reaches BLEU scores of 59.29 on the 'dev' set and 51.86 on the 'devtest' set after training for 150,000 steps and by ensembling the best 4 saved checkpoints. For our DE→HSB model, we obtain 57.22 on the 'dev' set and 49.95 on the 'devtest' set after training for 85,000 steps and by ensembling the best 4 saved checkpoints.

After the submission to the 2021 shared task, we continued training the above HSB→DE model up to 300,000 steps- Ensembling the *last* 4 saved checkpoints, BLEU scores were close to the ones shown in the last line of Table 3, reaching 59.42 on the 'dev' set and 51.37 on the 'devtest' set. However, several checkpoints gained almost 2 BLEU points on 'dev', pointing to the potential benefits of training for a longer time.

5 Baseline for Unsupervised DE→DSB Translation

Moreover, we studied the same techniques for translating Lower Sorbian (DSB), for which no parallel resources are provided. We translated the monolingual DSB data provided by the organizers with our HSB→DE model, hypothesizing that the differences between DSB and HSB are small enough to obtain an acceptable DSB-DE pseudo-parallel corpus, with high-quality text on the DSB side, following insights from our experience with Swiss-German dialects (Honnet et al., 2018).

We use the parameters from our best DE→HSB model to initialize a DE→DSB model that we train for 120k steps with the DSB-DE pseudo-parallel data. When ensembling the best 4 checkpoints, we reach BLEU scores of 8.25 / 8.22 without observing any significant increase of the scores during training. In fact, the initial score, which is the performance of a DE→HSB model on the DE-DSB 'devtest' data, is even slightly higher. An even lower BLEU score was reached when using our CS→DE model to translate monolingual DSB data into DE to obtain a pseudo-parallel corpus, thus confirming the finding that this approach does not lead to pseudo-parallel corpora of sufficient quality. Therefore, we did not submit these translations to the 2021 shared task.

6 Contrastive HSB↔DE and DE→DSB Systems using Multi-Task Learning

In contrast to the baseline systems presented above, we study an innovative approach, in which we train multitask systems with denoising auxiliary tasks that are presented in order of increasing complexity. This insight is drawn from curriculum learning (Bengio et al., 2009). We thus test whether increasing the complexity of the tasks makes it easier for an NMT model to learn the simple tasks first, and the harder ones later in training.

As Raffel et al. (2020) showed, source–to–source pre-training and multitasking improves translation, but not enough to compete with state-of-the-art setups. Therefore, instead, we perform target–to–target and source+target–to–target denoising. Considering their findings, we decide not to introduce special tokens into our vocabulary, such as mask tokens (instead just deleting the tokens with wish to mask), or sentence and language separators. Finally, due to computational constraints, we use the Transformer-Base as our architecture.

6.1 Data and Auxiliary Tasks

For our contrastive system we consider two new monolingual corpora in Czech and in German: the document-separated news crawls from WMT20 (Barrault et al., 2020), consisting of text extracted from online newspapers. They contain 17M lines and 43M lines respectively in each language. To keep training time within acceptable limits, we sample 1.4M lines from these corpora (including empty lines that serve as document-separators), we apply the same length-based filtering criterion ($2 < L < 301$) as for our baseline data, and we also delete all sentences that are made of more than 15% non-alphabetic characters. The resulting Czech corpus is 1.3M lines and 131,644 documents long, and the German corpus is 1.2M lines and 130,891 documents long.

For our document-level denoising tasks, we first divide into "chunks" a tokenized document-separated corpus so that each chunk is no more than 500 subwords in length, made up of consecutive lines in the same document; we only select documents made of at least 3 sentences. In Table 4 we list all corpora that we use to create our auxiliary data, including monolingual corpora back-translated with our baseline systems. The DE→DSB back-translated data was obtained with a baseline DE→HSB model.

We make use of the four following auxiliary denoising tasks (the main task being of course standard sentence-level translation, with all parallel and back-translated data), with the first two inspired by Devlin et al. (2019); Raffel et al. (2020) and Conneau and Lample (2019):

1. **Masking** (`MASK`): randomly delete 15% of words of a line on the source side, but keep the full original sequence on the target side.
2. **Translation Language Modeling** (`TLM`): concatenate the source and target sentences from a parallel corpus, and apply separately the `MASK` algorithm to each one. The target is the original target sentence.
3. **Mask Document First Words** (`MF`): for each chunk, leave the first sentence untouched, and for the remaining ones delete the first word of each sentence, with the target being the full original sequence in the same language.
4. **Next Sentence Generation** (`NSG`): for each chunk, leave all the sentences untouched except the last one, of which delete all but the two longest words; the model has to output the full original sequence. Keeping the two longest words (in characters) is based on the assumption that they are the most informative ones in the sentence.

The denoising tasks are listed above by increasing complexity. Indeed, `MASK`, as a monolingual sentence-level task, is the simplest denoising task we present, with `TLM` following, as it includes a context in a different language which needs to be identified. The two document-level tasks are more complex, as they require a larger context. In particular, NSG is harder than MF, since it consists of reconstructing a whole sentence with just two words from the original sequence, forcing the model to look for a more abundant context to estimate the correct answer. Furthermore, predicting the first word requires to take into account exclusively inter-sentential context, whereas masking a single random word allows also for the use of intra-sentential context, with the latter providing more direct context than the former.

Corpus	Lines	Words	Aux. tasks
CS-DE	1.5M	25M / 28M	TLM
HSB-DE	144k	2M / 2M	TLM
CS	1.3M	41M	MF, NSG
DE	1.2M	44M	MF, NSG
HSB	640k	9M	MASK
DSB	128k	2M	MASK
HSB→DE	4.5M	94M / 104M	
DE→HSB	637k	10M / 9M	
DE→DSB	124k	2M / 2M	

Table 4: Parallel (2), monolingual (4), and back-translated corpora (3) used for our contrastive system trained with multi-tasking. Each corpus is assembled from the raw datasets presented in Table 1 with the filtering setup described in Subsection 6.1. For bilingual corpora, we indicate the number of words in each language.

6.2 Training Schedules

All our models translate to one target language only, therefore the target side of our datasets is always the same language, be it for the monolingual denoising tasks or for TLM. Since all datasets correspond to sequence-to-sequence tasks, we are in essence simply removing and introducing datasets during training. The specific splits of the tasks in each training schedule have been manually set, guided by the reasons given below, without any attempt for fine-tuning.

All the hyperparameters of the models are those presented in Section 3, with the only exception of the parameters of CS↔DE models for initialization, which were trained on 4 GPUs to reduce training time. When we introduce new tasks during the training of a model, we continue training from the last checkpoint of the previous task.

Training CS↔DE models. Both directions are trained according to the same schedule, shown in Table 5, with simply the source and target languages switched. First, we train for 30k steps with a TLM task, then we train for another 30k steps with a mixture of the MF auxiliary task (50% of the samples) and the main translation task (50%). Then we continue for another 30k steps, changing MF to NSG. Finally, we finish with 30k steps on translation only. In total, the model is being trained for 30k steps (25%) with TLM, 15k steps (12.5%) with MF, 15k steps (12.5%) with NSG, and 60k steps (50%) with the main task, i.e. sentence-level translation.

	Steps ×1000			
Task	0-30	30-60	60-90	90-120
TLM	100%			
MF		50%		
NSG			50%	
Translation		50%	50%	100%

Table 5: Training schedule of the parent models in CS↔DE. For each direction, the model is only trained to output target language, so corpora differ depending on the direction (see 6.1). Both models are trained for 120k steps with three auxiliary denoising tasks and the main sentence-level translation task.

HSB→DE. The schedules of the child models are shown in Table 6 for the (DE, HSB) pair. For HSB→DE, we continue training from the best scoring checkpoint of the last 60k steps of the parent CS→DE model, and start with a TLM task for 60k steps. Then, we introduce back-translated data only for 60k steps. We continue with 60k steps with true parallel data only.

Additionally, we train two more models by continuing to train another 60k steps from the best scoring checkpoint (which is also the last one saved), with one of the models having its learning rate schedule reset. Although at first performance worsens due to a more aggressive learning rate during the warmup steps, the model ends up converging to a score similar to the one we obtain if we continue to train without resetting the learning rate schedule. The goal is to emulate a multiple-run seeding strategy for ensembling, by achieving a different weight distribution among the two models. We additionally train a randomly-initialized model with parallel data only, for 60k steps, also for ensembling. We generate our translations of the test data with an ensembling of 16 models: the best 4 checkpoints from the parallel-only randomly-initialized model, the best 4 of our main setup during the first 60k steps of parallel-only training, and the 4 checkpoints each for the two runs that continued to train with, and respectively without, resetting the learning rate schedule.

DE→HSB. We continue training from the best-scoring checkpoint of the last 60k steps of DE→CS, and provide it with a MASK task for 60k steps, since the model has not seen the target language at all during pre-training, for this direction. Then, we provide the model with a TLM task for 60k steps. Since in this direction we have much less back-translated data than in the opposite, we decide to train for 60k more steps with 50% of the samples being from the back-translated data, and the other 50% from the true parallel corpora. Finally, we continue training two more models in the same manner as explained for the HSB→DE direction. We additionally train a randomly-initialized parallel data only model for 60k steps for ensembling. We translate with the same ensembling setup as described for the HSB→DE direction.

	Steps ×1000		
Task	0-60	60-120	120-180
	HSB→DE		
TLM	100%		
Trans-BT		100%	
Trans-Parallel			100%
	DE→HSB		
MASK	100%		
TLM		100%	
Trans-BT			50%
Trans-Parallel			50%

Table 6: Training schedule of the child models for the HSB→DE and DE→HSB models presented in 6.2.

DE→DSB. We start training with a MASK task for 60k steps from the highest-scoring checkpoint DE→HSB. We continue training for 60k steps with just the back-translated data, although we notice that the quality of the translation affects negatively the scores. To address this issue, for another 60k steps we give it the back-translated corpus for 50% of the samples and the MASK task for the other 50%, starting training from the previous highest-scoring checkpoint. Finally, for another 60k steps we give it a parallel-only DE-HSB task for 50% of the samples, MASK for 30%, and back-translated data for 20%. After testing, using just the highest-scoring checkpoint for the back-translation only, back-translation + MASK, and DE-HSB + back-translation + MASK appeared to work better on the development data than using the highest four ones.

	Steps ×1000			
Task	0- -60	60- -120	120- -180	180- -240
MASK	100%		50%	30%
Trans-BT		100%	50%	20%
DE-HSB				50%

Table 7: Training schedule of the child DE→DSB models presented in 6.2

6.3 Results

The scores of the parent DE→CS and CS→DE models obtained with multi-task training are shown in Table 8. Compared to the CS→DE models from Sections 3 and 4, the present models have markedly lower scores. This difference can be due to the use of Transformer-Base vs. Big, or to differences in training data, apart from the multi-task training procedure itself. Still, we decided to use these models as parents for initializing the DE→HSB and HSB→DE models respectively, so that both parents and children are trained with multi-tasking. Although changes in the parameters of a parent model that result in better translations may not necessarily also result in better child initialization, it would be interesting to also test here the parent models from Section 4.

System	**DE→CS**	**CS→DE**
1. MF + translation	14.05	15.46
2. NSG + translation	15.30	16.17
3. Translation	18.19	19.80

Table 8: BLEU scores of parent models after each stage of the training schedule described in 6.2, on the 'devtest' set from 4.

Our child DE↔HSB models show that the scheduled training improves results over the baseline. The HSB→DE model with a training schedule (system 2 in Table 9), trained with a lighter architecture (Base vs. Big) and lower quality parent model (19.8 vs. 24.5), achieves a higher BLEU score than the system in Section 4, as shown in Table 3: 52.2 vs. 51.86. Additionally, the diversity of the ensembling of the models appears to improve the overall quality of the translation.

System	DE→HSB	HSB→DE
1. Parallel data	50.37	48.50
2. Multi-task	52.10	52.21
3. #2 cont. train	53.42	52.37
4. #2 cont. train with l. r. reset	53.05	52.12
Ensemble	54.58	53.21

Table 9: BLEU scores of child DE↔HSB models for various training schedules on the 2021 'devtest' set.

The scores of our DE→DSB model (Table 10) show that the quality of the back-translated data with our HSB→DE model improved slightly with the addition of the MASK monolingual task, but not with the addition of a DE→HSB translation task. However, when including in the ensemble the models trained on a DE→HSB task, scores improved from 8.7 to 9.6 on the 'devtest' set. This was the version submitted to the shared task on unsupervised MT (DE→DSB).

System	DE→DSB
1. Back-translation only	8.23
2. BT + MASK	8.57
3. BT + MASK + DE→HSB	7.14
Ensemble	9.62

Table 10: BLEU scores of child DE↔DSB models for various training schedules on the 2021 'devtest' set.

Finally, as we can see in Table 11, even with our possibly suboptimally trained parent models and lighter architecture, the strategy of diverse ensembles and scheduled multi-task training improved over our best performing baselines given in Section 4 for all directions of the low-resource MT task.

HSB→DE		DE→HSB		DE→DSB	
dev	devtest	dev	devtest	dev	devtest
62.74 *(+3.45)*	53.21 *(+1.35)*	62.49 *(+5.27)*	54.58 *(+4.63)*	9.22 *(+0.97)*	9.62 *(+1.40)*

Table 11: BLEU scores of our primary system's final configurations, on the development data, with the improvements over our highest baselines from Section 4.

7 Conclusion

In this work, we showed that non-iterative back-translation and parent-model transfer learning provide improvements for translation in a low-resource setting. Furthermore, multi-task scheduled training with monolingual or cross-lingual tasks also resulted in better models. In particular, child models starting with Translation Language Modeling tasks and Masking tasks improved over the baseline in all translation directions. Finally, our strategy of ensembling diverse models also produced higher scores than a mere checkpoint ensemble strategy.

Acknowledgments

We are grateful for their support to Armasuisse through the FamilyMT project, and to the Swiss National Science Foundation through grant n. 175693 for the DOMAT project: "On-demand Knowledge for Document-level Machine Translation".

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.

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 Conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

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.

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.

Pierre-Edouard Honnet, Andrei Popescu-Belis, Claudiu Musat, and Michael Baeriswyl. 2018. Machine translation of low-resource spoken dialects: Strategies for normalizing Swiss German. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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. *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Jindřich Libovický, Viktor Hangya, Helmut Schmid, and Alexander Fraser. 2020. The LMU Munich system for the WMT20 very low resource supervised MT task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1104–1111, Online. 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.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

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. 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.

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.

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.