Samsung Research Philippines - Datasaur AI's Submission for the WMT22 Large Scale Multilingual Translation Task

Jan Christian Blaise Cruz
Samsung Research Philippines
Manila, Philippines
jcb.cruz@samsung.com

Lintang Sutawika[*]
Datasaur AI
San Francisco, California, USA
lintang@datasaur.ai

Abstract

This paper describes the submission of the joint Samsung Research Philippines - Datasaur AI team for the WMT22 Large Scale Multilingual African Translation shared task. We approach the contest as a way to explore task composition as a solution for low-resource multilingual translation, using adapter fusion to combine multiple task adapters that learn subsets of the total translation pairs. Our final model shows performance improvements in 32 out of the 44 translation directions that we participate in when compared to a single model system trained on multiple directions at once.

1 Introduction

In this paper, we describe two systems that we submit to the WMT22 Large Scale Multilingual African Translation shared task: a baseline finetuned MT5 (Xue et al., 2020) model trained on multiple directions at once (referred to as `SRPH-DAI-Baseline`), and an MT5 model successively finetuned with task composition using multiple pair-specific adapters (referred to as `SRPH-DAI-Fusion`).

We first outline the preprocessing steps and filtering heuristics used to clean the contest dataset, then we show the training setup and experimental design used for constructing our submitted systems. We then report our results on the hidden test set via BLEU, spBLEU, and CHRF2++ automatic evaluation metrics.

2 Preprocessing

In this section, we detail the preprocessing steps used to filter the contest dataset to ensure that data quality is as high as possible.

Given that the contest dataset contains sentence pairs that were artificially aligned from crawled data, we use a number of filters to reduce the possibility of mismatched pairs in the final training dataset:

- We filter out pairs where one or both sentences have too few ($<= 3$) or too many ($>= 150$) tokens post-sentencepiece tokenization.
- We remove pairs if one or both sentences have too many repeated ($>= 5$) punctuations or symbols of the same type (e.g. "/////"), or contiguous punctuations/symbols of considerable ($>= 3$) length (e.g. "`word` $&**$").
- We also remove sentence pairs where one sentence has punctuation that is missing from the other (e.g. "`word`!!" → "`word`?").
- If a pair has a sentence where a large percentage of the total characters (total $>= 70\%$) are numbers or punctuations (e.g. "`word` ??! +22 8456 8967"), the pair is dropped.
- An average word length filter is also used to remove pairs where one sentence has words that are disproportionately longer than the words in the corresponding sentence. We get a ratio r by taking the sum of the lengths of each token in a sentence, then dividing it by the number of tokens. We only keep sentence pairs where both sentences have a ratio r within $3 <= r <= 15$.
- HTML and URL-containing sentence pairs are also removed as this contributes to unnecessary noise during training.
- Lastly, we also check for known word matches within each sentence pair. For instance, if we detect a number (e.g. "1" or "one"), we also check the corresponding sentence for the same number. Sentence pairs that have mismatched

[*] Work done while at Konvergen AI.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1034–1038
Abu Dhabi, December 7–8, 2022. ©2022 Association for Computational Linguistics

Pair	Samples
afr ↔ eng	2,526,513
amh ↔ eng	315,870
fuv ↔ eng	953,002
hau ↔ eng	1,841,974
ibo ↔ eng	136,534
kam ↔ eng	1,143,082
kin ↔ eng	7,143,167
lug ↔ eng	2,058,590
luo ↔ eng	1,713,159
nso ↔ eng	1,600,977
nya ↔ eng	1,289,859
orm ↔ eng	1,786,712
sna ↔ eng	5,917,741
som ↔ eng	413,647
ssw ↔ eng	77,807
swh ↔ eng	18,243,580
tsn ↔ eng	3,034,232
tso ↔ eng	383,586
umb ↔ eng	190,170
xho ↔ eng	5,481,855
yor ↔ eng	923,055
zul ↔ eng	2,645,396

Table 1: Final dataset statistics after running the sentence pair filters.

(e.g. source sentence has “1” but target sentence has “11”) words are dropped as these are likely from misaligned data.

After applying the filters for the entire dataset, we perform one deduplication step to ensure that no duplicate entries have been added. No further preprocessing is done on the data itself to preserve as much information within the sentences as possible.

When formatting the data for translation training, we insert a target language token at the beginning of the sentence. For example, a sentence to be translated from English to Afrikaans would look like:

`<afr>` This is an example sentence.

We only participate in a subset of the shared task's translation pairs (44 total directions), opting to train only on English → African and African → English pairs due to resource constraints.

3 Experiment Design

In this section, we describe the construction of our two submitted systems: `SRPH-DAI-Baseline` and `SRPH-DAI-Fusion`.

3.1 Common Settings

Both systems use MT5-Small, a Transformer-based (Vaswani et al., 2017) model, as an initialization point. We opted to use the small variant (∼300 million parameters) as opposed to the bigger base (∼580 million) and large (∼1.2 billion) variants due to resource constraints in our setup. We expect the performance of our models to further improve as we scale to larger variants of pretrained models.

As a remedy to constrained resources as well as a way to improve stability during training for low-resource data, we decided to use adapters (Houlsby et al., 2019; Pfeiffer et al., 2020b) instead of fully finetuning all the model parameters.

Before proceeding to training for translation, we first train a language adapter (Pfeiffer et al., 2020b) on English + African languages in order to better condition the MT5 model for the languages it will encounter later. We mimic MT5's pretraining and use span corruption on the provided monolingual training data for the shared task (which is likewise filtered like our parallel data).

We freeze the pretrained weights and train the adapter for a total of 150K steps using the Adafactor (Shazeer and Stern, 2018) optimizer, utilizing a learning rate schedule that warms up for the first 10K steps to a maximum of $1e-4$, then linearly decaying after. We use a maximum sequence length of 512 for language adapter training, using gradient accumulation to train with a total batch size of 128 sequences per training step. The output language adapter is used in both of our submission systems, and is stacked below the translation task adapter(s).

3.2 Baseline Model

We construct our baseline model `SRPH-DAI-Baseline` by stacking a blank task adapter on top of our language adapter and training it on all 44 translation directions at once. In this setup, the language adapter is frozen. This model is trained for a total of 300K steps on the combined filtered dataset using the Adafactor optimizer. We use a learning rate of $5e-5$ and a weight decay of $1e-8$, warming up for the first 10K steps, then linearly decaying after.

Unlike other systems, we do not perform any other techniques such as backtranslation (Edunov et al., 2018), noisy channel reranking (Yee et al., 2019), or clever pair sampling (Fan et al., 2021) in order to further boost performance. This mimics an "ablation" setup where only the direct finetuning method is used in order to accurately observe the effect of using task composition later on. Since no further modifications are made on the model beyond the training method, any improvements on performance made by task composition can be attributed to task composition and not anything else.

3.3 Exploring Task Composition

In the conventional multidirectional setup like in our baseline, the model learns generic cross-lingual information at the same time that it learns task-specific information. Learning cross-lingual information is useful in cases where a number of the languages in the model are similar or come from the same family (Saleh et al., 2021; Siddhant et al., 2022). However, in cases where a number of the languages are dissimilar or come from different families, we hypothesize that it may be useful to learn cross-lingual information *separately* from task-specific mappings. This ensures that the model learns each translation direction in a non-destructive manner with respect to other language pairs.

In cases where certain language pairs are underrepresented in the training set, learning each direction separately also removes the need for specialized data sampling methods to ensure that the model sees each pair enough times. In addition, using adapters for low-resource pairs also helps prevent overfitting the small dataset (Mao et al., 2021).

Motivated by this, instead of finetuning a task adapter for multiple translation directions, we instead opt to train *multiple* translation task adapters to learn task-specific information, then *composing* the multidirectional setup afterwards via Adapter Fusion (Pfeiffer et al., 2020a) to mix cross-lingual and cross-task information. This is how we construct our `SRPH-DAI-Fusion` model.

For this setup, we follow the same training routine as in the baseline, except we only train on one language pair at a time. We train an adapter to produce translations for *two* directions: English $\rightarrow X$ and $X \rightarrow$ English. Training in more than one direction ensures that the task adapters learn to properly embed the target language token at the beginning of every sentence. This results in a total of 22 task adapters for each of the 22 English $\rightarrow X$ pairs.

Finally, we add an Adapter Fusion setup for all 22 single-pair task adapters, freeze the adapters,then further finetune the model to learn cross-task and cross-lingual information. We finetune for 100K steps with a learning rate of $2e-5$ using the Adafactor optimizer. Like in previous setups, we also use a warmup of 10K steps with a linear decay afterwards.

4 Results

We outline the performance of our two models on the hidden test set on Table 2.

Overall, `SRPH-DAI-Fusion` outperforms `SRPH-DAI-Base` on average across all three metrics, with an improvement of 0.09, 0.19, and 1.33 on average BLEU, spBLEU, and CHRF2++, respectively. Both models perform relatively better on the African to English translation directions compared to the English to African ones. We hypothesize that this is likely due to English being a pivot language, and thus cross-lingual and cross-task information learned while training each pair contributed to better performance when translating into English.

When comparing the two models, we note an "improvement" in the performance if at least two of the three metrics had an increase in score. We observe that 32 out of the 44 translation directions had an improvement once task composition was used for finetuning, most of which are very low-resource pairs. Best gains are observed in the English to African translation directions, with some pairs such as Eng $\rightarrow$ Orm improving from an initial 0 score from the baseline model.

We observe that `SRPH-DAI-Base` outperforms the task composition model in cases where there is a relative abundance of training data. For pairs that have sub-million examples, `SRPH-DAI-Fusion` performs much better, likely due to the model being able to learn more specialized information about these translation directions separate from the other directions.

Interestingly, we observe that for language pairs with a relative abundance of data, the drop in performance when using task composition is substantial. For example, Afr $\rightarrow$ Eng suffers a 2.2, 2, and 4.2 points drop in BLEU, spBLEU, and CHRF2++, respectively. We hypothesize that this is because `SRPH-DAI-Fusion` has more intact task-specific knowledge related to low-resource pairs that may not be useful to the higher-resourced pairs. Since task adapters are frozen during fusion layer training, the model has an added burden in learning how to adapt knowledge that may not be useful when translating higher-resourced translation directions.

5 Conclusion

In this paper, we described our submissions for the WMT22 Large Scale Multilingual African Translation shared task. We approached the contest as a way to explore task composition as a solution for multilingual translation, especially among low-resource languages. In our experiments, we show that using task composition – training task adapters to learn pair-specific knowledge, then using a fusion layer to learn cross-task information – improves performance for less-represented language pairs in a multilingual translation dataset. While the model's results for a number of translation directions are far from state-of-the-art, the results show the methodology's promise for further exploration.

For future work, we would like to conduct experiments for larger models than is constrained by our resources. We expect that using Base and Large variants of MT5 would further improve performance for all language pairs. In addition, it would be beneficial to test the methodology while adding in common "best practices" in translation such as using backtranslated data and better data sampling. Lastly, we would like to explore setups where the pair-specific task adapters are *transformable* to some extent instead of being fully frozen as a remedy to the problem of higher-resourced pairs performing worse in the task composition setup.

References

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. 2021. Unipelt: A unified framework for parameter-efficient language model tuning. *arXiv preprint arXiv:2110.07577*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Fahimeh Saleh, Wray Buntine, Gholamreza Haffari, and Lan Du. 2021. Multilingual neural machine translation: Can linguistic hierarchies help? *arXiv preprint arXiv:2110.07816*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Kyra Yee, Nathan Ng, Yann N Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. *arXiv preprint arXiv:1908.05731*.

	SRPH-DAI-Base			SRPH-DAI-Fusion			
Pair	**BLEU**	**spBLEU**	**CHRF2++**	**BLEU**	**spBLEU**	**CHRF2++**	**Improved?**
afr → eng	8.3	9.1	26.2	6.1	7.1	22	-
amh → eng	0.7	0.8	11.2	0.9	1	11.3	✓
fuv → eng	1.3	1.8	10.6	1.4	2	11.1	✓
hau → eng	2.7	3.7	14.8	2.5	3.6	14.6	-
ibo → eng	1.9	2.6	12.3	2.1	3	12.7	✓
kam → eng	2.1	2.8	12.1	2.1	2.9	12.6	✓
kin → eng	2.3	3.1	14.2	2.7	3.4	15	✓
lug → eng	1.8	2.4	11.9	2	2.6	13	✓
luo → eng	1.8	2.2	11	1.8	2.4	11.7	✓
nso → eng	2.8	3.6	14.3	3.1	4.2	15.9	✓
nya → eng	3	3.9	15.4	3.1	4.2	15.9	✓
orm → eng	0.5	0.7	8.4	0.6	0.9	9.2	✓
sna → eng	3	3.7	15.1	3	3.7	15.5	-
som → eng	2	2.5	12.4	2.3	3	14	✓
ssw → eng	2.6	3.3	13.8	2.6	3.3	14	✓
swh → eng	4.1	4.4	18.1	3.9	4.6	17.8	-
tsn → eng	2.3	2.9	13.3	2.6	3.3	14.1	✓
tso → eng	2.1	2.8	12.3	2.4	3	13.1	✓
umb → eng	1	1.5	10.7	0.9	1.5	11	-
xho → eng	3.3	4.1	16.4	3.2	4	16.3	-
yor → eng	1.5	2.1	10.9	1.8	2.5	12.2	✓
zul → eng	2.9	3.5	15.4	2.9	3.5	15.4	-
eng → afr	4.1	4.3	20.6	2.6	3	17.9	-
eng → amh	0.1	0	2.6	0.3	0.2	0.5	✓
eng → fuv	0.1	0.1	4	0.9	1.2	10	✓
eng → hau	0.3	0.5	8.3	0.5	1	19.4	✓
eng → ibo	0.2	0.1	4.4	0.6	0.8	8.7	✓
eng → kam	0.1	0.1	2.5	0.6	0.8	7.9	✓
eng → kin	0.3	0.4	5.4	0.4	0.4	8.1	✓
eng → lug	0.2	0.3	3.8	1.1	0.9	8.7	✓
eng → luo	0.5	0.6	5.7	1	1.4	10.1	✓
eng → nso	0.3	0.4	5.5	0.4	0.9	8.3	✓
eng → nya	1.1	0.8	10.6	1.4	1.4	11.5	✓
eng → orm	0	0	2.2	0.1	0.1	4.7	✓
eng → sna	1.1	0.8	11.8	1	0.8	9.3	-
eng → som	0.3	0.1	6.8	0.4	0.5	7.4	✓
eng → ssw	0.7	0.8	9	1.1	0.8	9.2	✓
eng → swh	1.3	1.4	14.9	1.1	1.6	13.2	-
eng → tsn	0.3	0.3	5.6	0.3	0.7	7.7	✓
eng → tso	0.3	0.4	3.8	0.7	1.1	8.9	✓
eng → umb	0.3	0.2	3.1	0.6	0.7	8.3	✓
eng → xho	0.6	0.9	11.1	0.6	0.6	11	-
eng → yor	0.1	0	4.1	0.3	0.3	6.3	✓
eng → zul	0.5	0.8	10.6	0.6	0.5	10.1	-
Average	1.52	1.84	10.39	1.61	2.03	11.72	

Table 2: Results of both SRPH-DAI-Base and SRPH-DAI-Fusion on the hidden test set. We consider task composition as an improvement if it resulted in an increase in performance in *at least two* of the three automatic metrics.