

LEARNING WITHOUT FORGETTING: TASK AWARE MULTITASK LEARNING FOR MULTI-MODALITY TASKS

Anonymous authors

Paper under double-blind review

1 APPENDIX

1.1 DATASETS

1.1.1 DATA AUGMENTATION FOR SPEECH TRANSLATION

Table 1 provides details about the datasets used for the multi-modality experiments. Since En-De ST task has relatively fewer training examples compared to ASR and MT tasks, we augment the ST dataset with synthetic training examples. We generate the synthetic speech sequence and pair it with the synthetic German text sequences, obtained by using the top two beam search results of the two trained English-to-German NMT models. For speech sequence, we use the Sox library to generate the speech signal using different values of speed, echo, and tempo parameters similar to (Potapczyk et al., 2019). The parameter values are uniformly sampled using these ranges : tempo $\in (0.85, 1.3)$, speed $\in (0.95, 1.05)$, echo_delay $\in (20, 200)$, and echo_decay $\in (0.05, 0.2)$. We also train two NMT models on EN-De language pair to generate synthetic German sequence. The first model is based on Edunov et al. (2018) and the second model (Indurthi et al., 2019) is trained on WMT’18 En-De and *OpenSubtitles* datasets. We increase the size of the IWSLT 19(filtered) ST dataset to five times of the original size by augmenting 4x data – four text sequences using the top two beam results from each EN-De NMT model and four speech signals using the Sox parameter ranges. For the Europarl-ST, we augment 2x examples to triple the size. The TED-LIUM 3 dataset does not contain speech-to-text translation examples originally; hence, we create 2x synthetic speech-to-text translations using speech-to-text transcripts. Finally, for the MuST-C dataset, we only create synthetic speech and pair it with the original translation to increase the dataset size to 4x. The Overall, we created the synthetic training data of size approximately equal to four times of the original data for the ST task.

1.1.2 TASK IDENTIFICATION WITHOUT TASK INFORMATION

Under the multi-modality setting, we conducted smaller scale experiments using only one dataset for each ST, ASR, and MT tasks. The details of the datasets used have been provided in Table 3. We trained on single p40 GPU for 400k steps. The corresponding results have been reported in Table 2. All the results have been obtained without any finetuning. Even though our task-aware MTL model achieves significant performance improvement over vanilla MTL models, we can observe that the vanilla MTL models are also able to give a decent performance on all tasks without any finetuning. An explanation for this is that we used MuST-C dataset for the En-De ST task and TEDLIUM v3 for the ASR task, which means that the source speech is coming from 2 different sources. However, if we use the same datasets for both the tasks(after data augmentation), the MTL models get confused and the ST, ASR outputs are mixed. The MTL models might be able to learn the task identities simply based on the source speech sequences, since these sequence are coming from different datasets for each task type–MuST-C for ST and TED-LIUM v3 for ASR. However, this does not mean that vanilla MTL models perform joint learning effectively. A human who can perform multiple tasks from the same input is aware of the task he has to perform beforehand. Similarly, it is unreasonable to expect different outputs (translation, transcription) from a model to the same type of input (English speech) without any explicit task information.

Task	Corpus	# hours	# Examples
MT	Open Subtitles	N/A	22,512,639
MT	WMT 19	N/A	4,592,289
ASR	LibriSpeech	982	232,958
ASR	IWSLT 19 ST(filtered)	220	145,372
ASR	MuST-C	400	229,702
ASR	CommonVoice	1469	232,958
ASR	TED-LIUM 3	452	286,263
ST	Europarl-ST	89	32,628
ST	IWSLT 19 ST(filtered)	220	145,372
ST	MuST-C	400	229,703

Table 1: Number of original training examples in each dataset.

S No.	MTL Strategy	MT BLEU (\uparrow)	ASR(WER (\downarrow))		ST(BLEU (\uparrow))	
		Test	Dev	Test	Dev	Test
1	Joint Learning	14.77	29.56	30.87	13.10	12.70
2	Meta Learning	14.74	28.58	29.92	13.89	13.67
This Work						
3	Task Aware Meta Learning (with implicit TCN)	18.84	21.29	23.44	17.77	17.51

Table 2: Performance of models trained using different approaches on the ASR, MT and ST tasks using different datasets

1.1.3 IMPLEMENTATION DETAILS

The detailed hyperparameters settings used for the single modality and multi modality experiments have been provided in the Table 4.

REFERENCES

- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1045. URL <https://www.aclweb.org/anthology/D18-1045>.
- Sathish Reddy Indurthi, Insoo Chung, and Sangha Kim. Look harder: A neural machine translation model with hard attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3037–3043, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1290. URL <https://www.aclweb.org/anthology/P19-1290>.
- Tomasz Potapczyk, Pawel Przybysz, Marcin Chochowski, and Artur Szumaczuk. Samsung’s system for the iwslt 2019 end-to-end speech translation task. In *16th International Workshop on Spoken Language Translation (IWSLT)*. Zenodo, November 2019. doi: 10.5281/zenodo.3525498. URL <https://doi.org/10.5281/zenodo.3525498>.

Task	Corpus	Train		Dev	Test
		hours	Examples	Examples	Examples
MT	WMT 14	N/A	4,592,289	3,000	3,003
ASR	TED-LIUM 3	452	286,263	1,469	591
ST	MuST-C	400	229,703	1,423	2,641
	Synthetic	N/A	689, 103	N/A	N/A

Table 3: The data statistics of ASR, MT and ST tasks used in our experiments.

Hyperparameter	<i>Single Modality</i>	<i>Multi Modality</i>
batching	dynamic	static
batch size	2048 (tokens)	16 (examples)
optimizer	adam	adam
adam_betas	(0.9,0.997)	(0.9,0.997)
lr_scheduler	inverse_sqrt	inverse_sqrt
lr	2.0	2.0
lr_warmup_steps	16000	16000
label_smoothing	0.1	0.1
dropout	0.1	0.1
lr_decay_rate	1.0	1.0
hidden_size	512	512
encoder_layers	6	12
decoder_embed_dim	512	512
decoder_layers	6	12
num_heads	8	8
filter_size(ffn layers)	1024	1024

Table 4: Hyperparameter details for the experiments