POSTECH Submission on Duolingo Shared Task

Junsu Park[1]**, Hongseok Kwon**[1]**, Jong-Hyeok Lee**[1,2]
Department of Computer Science and Engineering[1], Graduate School of Artificial Intelligence[2]
Pohang University of Science and Technology (POSTECH), Republic of Korea
{jspak3, hkwon, jhlee}@postech.ac.kr

Abstract

This paper describes POSTECH's submission to the 2020 Duolingo Shared Task on Simultaneous Translation And Paraphrase for Langauge Education (STAPLE) for the English-Korean language pair. In this paper, we propose a transfer learning based simultaneous translation model by extending BART. We pre-trained BART with Korean Wikipedia and a Korean news dataset, and fine-tuned it with an additional web-crawled parallel corpus and the 2020 Duolingo official training dataset. In our experiments on the 2020 Duolingo test dataset, our submission achieves 0.312 in weighted macro F1 score, and ranks second among the submitted En-Ko systems.

1 Introduction

Simultaneous Translation And Paraphrase for Language Education (STAPLE) is the task of automatically producing multiple translations from a single source sentence (Mayhew et al., 2020). Because STAPLE can be regarded as a mixture of the machine translation (MT) and paraphrasing problem, MT and paraphrasing techniques play an important role in this task. Unlike in a typical MT task, systems are demanded to generate high-coverage sets on a sentence-level, as opposed to word-level. Subsequently, systems require a deeper linguistic understanding of the target language to generate accurate target sentences.

Recent NLP studies have alleviated this problem by transfer learning (Ventura and Warnick, 2007) from pre-trained language models. Radford et al. (2018) proposed a generative pre-trained language model (GPT), which trains a Transformer decoder with large-scale monolingual data, to achieve significantly improved performance in nine out of the twelve datasets. Despite these improvements, GPT shows a limited ability to model bidirectional context due to using the classical generative modeling approach. On the other hand, Devlin et al. (2018) proposed bidirectional encoder representations from Transformers (BERT), trained for the reconstruction of natural language from sentences containing masked tokens, in order to obtain deeper representations for natural language. By training on an enormous amount of training data, they achieved state-of-the-art results on eleven NLP tasks. To take advantage of both pre-trained generative models and pre-trained bidirectional encoders, Lewis et al. (2019) introduced a denoising autoencoder for pre-training sequence-to-sequence models called BART. BART aims to learn linguistic knowledge in the process of first corrupting the text using various noise functions and then restoring it, and showed state-of-the-art performance in various tasks.

Given this background, we expected that using a transfer-learning-based approach could resolve two difficulties of the En-Ko track of STAPLE: data insufficiency and multiple sentence generation. Unlike recent MT models which used over 4.5 million sentence pair for training data, the STAPLE official dataset includes only 2500 En-Ko source sentences. With such small data, we predicted that recent NMT models would not be able to learn translation knowledge effectively. Also, we speculated that paraphrasing requires a deep understanding of the language. Based on this prediction, a well-trained language model and a generative model for target language were needed to achieve this task's objectives.

With these considerations, we concluded that BART, a sequence-to-sequence generative model pre-trained on a large amount of data, is most suitable for STAPLE and thus propose a transfer-learning-based simultaneous translation model by extending BART. Our model added a randomly initialized source-side encoder in place of the embedding layer of BART pre-trained by Korean monolingual data and predicts translation weights with

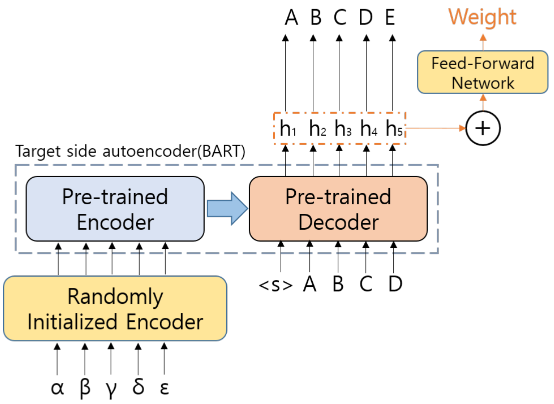

Figure 1: The overall architecture of the proposed model. The input vectors of feed-forward network are the sum of the pre-trained decoder's hidden vectors.

an additional feed-forward network using hidden vectors generated by the pre-trained decoder. The remainder of the paper is organized as follows: Section 2 describes our proposed method. Section 3 summarizes the experimental procedure and results, and Section 4 gives the conclusion.

2 Method

We adopt BART to the STAPLE problem, which takes source sentence to generate multiple target sentences. Our model consists of a pre-trained autoencoder with the source-side encoder that proposed in Lewis et al. (2019) and a feed-forward network to predict translation weights (Figure 1). In the following subsections, we describe our methods in detail.

2.1 Pre-trained autoencoder (BART)

We used BART as our pre-trained autoencoder structure. As was with BART, our autoencoder structure learns linguistic information of the target language by denoising various types of document corruptions. Among the five document corruption types proposed by BART, we applied Text Infilling and Sentence Permutation because they yielded the best results on Lewis et al. (2019).

2.2 Source-side Encoder

Pre-trained BART is a monolingual model, so the proposed model needs an additional encoder to function as translation model. After pre-training BART, we removed the embedding layer of the pre-trained encoder and added a randomly-initialized encoder instead (Lewis et al., 2019). In order to prevent corruption from the high loss in the randomly-initialized encoder during initial training, we freeze all pre-trained BART weights during the first fine-tuning step except for the self-attention input projection matrix of BART's first encoder layer. In the second step, we train all model parameters.

Dataset	**Sentence**	**Word**
Monolingual	31,654,593	447,754,804
Additional parallel	2,035,566	29,964,677
Official (1 to 1)	700,410	2,915,939

Table 1: **Dataset statistics** - number of target sentence and word.

2.3 Feed-forward network for translation weight training

We added a feed-forward network to predict a translation weight on each generated sentence. The sum of hidden vectors which generated on the decoder is passed as the input of the feed-forward network. The output of the feed-forward network passed through a sigmoid layer becomes the final translation weight. During the generation step, the sentences with the high weights are selected.

3 Experiments

3.1 Dataset

Pre-training. For pre-training, we use text crawled from the Korean Wikipedia (5.8M words) and Korean online news sites (447M words). When crawling, we extracted only text passages and ignored headers, lists, and tables. To reduce training time, we filtered out any samples that exceed 100 tokens.

Fine-tuning. For fine-tuning, we used the STAPLE official training data (Duolingo, 2020) (700K sentences), setting aside 100 sentences each for the development set and test set. In addition, we adopted the web crawling parallel corpus (2M sentences) as additional training and development data for the source-side encoder. As with the pre-training corpus, we filtered out any training or development samples longer than 100 tokens.

3.2 Training Details

Settings. We modified the Fairseq (Ott et al., 2019) implementation of BART to build our model. Most hyperparameters of BART pre-training such as dropout ratio, hidden size, and etc. were copied from the base model described in Lewis et al.

	Decoding Option			Weighted Macro F1 ↑	Weighted Recall ↑	Precision ↑
	Beam Size	Diverse	Nbest (weight)			
Beam search	50	–	50	0.3192	0.3092	**0.5202**
	75	–	**75**	**0.3280**	0.3651	0.4628
	100	–	100	0.3234	0.4008	0.4214
	140	–	140	0.3108	0.4394	0.3680
	500	–	500	0.2218	**0.5817**	0.1865
Diverse beam search	100	5	100	0.1673	0.2069	0.2212
	100	10	100	0.1164	0.1474	0.1601
Beam search with weight	75	–	50	0.2695	0.2546	0.4630
	75	–	65	0.3064	0.3197	0.4615
	75	–	70	0.3163	0.3410	0.4596

Table 2: **Results of training variants** – each separated section corresponds to a different generation strategy (Beam search, Diverse beam search and Beam search with weight). Diverse is the number of group for diverse beam search and Nbest (weight) is the number of sentences selected by highest translation weight. The bold values indicate the best result in the metrics for each architecture.

(2019). For the document corruption scheme, we used the pre-training options of Lewis et al. (2019): Text Infilling and Sentence Shuffling. We set warm-up learning steps to 10K out of 250K total steps. For data preprocessing, we applied the sentencepiece (Kudo and Richardson, 2018) implementation of byte-pair encoding (Sennrich et al., 2016) with a 32k vocabulary on each language.

Pre-training. We trained target-side BART using Text Infilling and Sentence Shuffling as described in §2.1. We replaced 30% of tokens with single [MASK] symbols with span length distribution ($\lambda = 3$) on Text Infilling.

Fine-tuning. We divided fine-tuning step into four steps.

1. **Pre-train source-side encoder** After pre-training, we detached the embedding layer of BART encoder and attached a randomly initialized encoder as described in §2.2. We used only our web crawling parallel corpus for this step. During this step, we freeze the pre-trained model except the first encoder layer's projection weights to prevent the pre-trained weights being affected by the high loss while the encoder learns the source-side representation.

2. **Fine-tuning on MT** After pre-training the source-side encoder, we trained entire model on the same training data with a smaller learning rate. Because the size of the parallel data used for fine-tuning is much smaller than that of monolingual data used for pre-training, we expected pre-trained BART to generate the correct sentences even if the source-side encoder produced an incorrect expression.

3. **Fine-tuning on paraphrasing** After training on an additional parallel corpus, we trained the entire model on the official parallel corpus to reach the paraphrasing goal.

4. **Weight training** After learning all sentence representations, we trained a feed-forward network for translation weight prediction on the official target language weights. In order to train translation weights without corrupting the sentence generation model, we freeze all parts of the model excluding the feed forward network.

Experiment variations. We conduct multiple experiments on test set divided from official training set to determine the best generation strategy.

- **Beam search** with different beam size. We selected all generated sentences.

- **Diverse beam search** with different beam size and group size. We used the implementation of Vijayakumar et al. (2016).

- **Beam search w/ weight** with same beam size but different size of sentences selected by highest translation weight.

Systems	Weighted Macro F1 ↑	Weighted Recall ↑	Precision ↑
jbrem	**0.4035**	**0.4518**	0.4795
jspak3 (ours)	0.3116	0.3342	0.4701
sweagraw	0.2553	0.3168	0.3216
jindra.helcl	0.2058	0.1935	0.3894
STAPLE_fairseq_baseline	0.0486	0.0315	0.2204
STAPLE_aws_baseline	0.0412	0.0226	**0.6360**

Table 3: **Submission results** – the official results of 2020 Duolingo shared task in En-Ko language pair. The bold values indicate the best result in the metrics for the each architecture.

3.3 Results

We trained the model as described in §3.2 using various generation strategies. For evaluation, we used weighted macro F1 scores on our test set extracted from the 2020 Duolingo official dataset. Table 2 shows the scores of each generation strategy. In the case of beam size, results showed the highest weighted macro F1 score when the beam size was 75. We speculate this to be because of the trade-off between weighted recall and precision. Using diverse beam search with beam size 100 and beam search with translation weight showed ineffective results. We initially expected to attain a higher precision with similar weighted recall if the translation weights were predicted accurately, but it seems our feed-forward network was not able to learn the distribution of translation weights properly. Also, we had expected diverse beam decoding to help generate more diverse sentences, but it had an adverse effect on overall performance.

Submission results. The submission results on the official test set are reported in Table 3. We selected the decoding option obtained by applying beam search with beam size 75, Nbest 75 which showed the highest weighted macro F1 score in Table 2 as our final submission. Our submission achieves an improvement of +0.263 in weighted macro F1 score compared to the baseline. As a result, our system ranks second out of the four systems submitted this year.

4 Conclusion

In this paper, we present POSTECH's submissions to the 2020 Duolingo shared task. We propose a transfer-learning based simultaneous translation model by extending BART. The proposed model is first pre-trained by reconstructing large corrupted text using text infilling and sentence shuffling, and then fine-tuned with an additional parallel corpus and the official training dataset with a newly added randomly initialized encoder in place of the embedding layer. It has an additional feed-forward network to predict translation weight trained on the official dataset. Finally, our model outperforms the baseline by a large margin and ranks second out of the submitted systems.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Duolingo. 2020. Data for the 2020 Duolingo Shared Task on Simultaneous Translation And Paraphrase for Language Education (STAPLE).

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

S. Mayhew, K. Bicknell, C. Brust, B. McDowell, W. Monroe, and B. Settles. 2020. Simultaneous translation and paraphrase for language education. In *Proceedings of the ACL Workshop on Neural Generation and Translation (WNGT)*. ACL.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL* *https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

Dan Ventura and Sean Warnick. 2007. A theoretical foundation for inductive transfer. *Brigham Young University, College of Physical and Mathematical Sciences*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models.