System Description for Transperfect

Wiktor Stribiżew and **Fred Bane** and **José Conceição** and **Anna Zaretskaya**
Transperfect Translations
{wstribizew, fbane, jconceicao, azaretskaya}@translations.com

Abstract

In this paper, we describe our participation in the 2021 Workshop on Asian Translation (team ID: tpt_wat). We submitted results for all six directions of the JPC2 patent task. As a first-time participant in the task, we attempted to identify a single configuration that provided the best overall results across all language pairs. All our submissions were created using single base transformer models, trained on only the task-specific data, using a consistent configuration of hyperparameters. In contrast to the uniformity of our methods, our results vary widely across the six language pairs.

1 Introduction

The field of machine translation has seen rapid innovation in the last few years, with new model architectures, pre-training regimens, and computational algorithms emerging at a dizzying pace. However, translation of these techniques into industry practice occurs more slowly. Companies utilizing these techniques must take into account considerations such as deployment costs (model speed and size), scalability, explainability, the complexity of training regimens (resource constraints limiting independent hyperparameter optimization for all language pairs), and risk management, against which advances yielding performance gains must be weighed.

For our participation in the 2021 Workshop on Asian Translation shared task on patent translation, we have applied a single, standardized data preparation and model training pipeline as a way of benchmarking the performance of this process. We conducted limited experiments to test different parameters, before settling on the approach which provided the best overall results across all language pairs. Our NMT systems are standard base Transformer (Vaswani et al., 2017) models, which were trained using only the data resources provided by the task organizers. These models used shared subword vocabularies created with SentencePiece (Kudo and Richardson, 2018).

In contrast to the uniformity of our methods, our results varied widely across the six language pairs. Different scoring metrics prevent the direct comparison of scores from different language pairs, but relative to the top performing model in each language pair, our scores ranged from 98.84% of the top score for the English → Japanese language pair, to 83.89% of the top score for Korean → Japanese. Below, we describe in detail our system architecture, hyperparameter configuration, hardware resources, and results.

2 System Overview

2.1 Task Description

The JPC2 patent task consisted of translation in the patent domain between English and Japanese, Korean and Japanese, and Chinese and Japanese. The training data consisted of parallel corpora provided by the Japan Patent Office (JPO), with training sets containing one million sentence pairs for each language pair. The data are drawn from four domains, chemistry, electricity, mechanical engineering, and physics.[1]

2.2 Data Processing

The data were encoded using subword encodings learned from the corpora using the unigram model trainer provided by SentencePiece (Kudo and Richardson, 2018). To avoid the added complexity of using different pre-tokenization strategies for

[1] http://lotus.kuee.kyoto-u.ac.jp/WAT/patent/

Proceedings of the 8th Workshop on Asian Translation, pages 138–140

different languages, we did not pre-tokenize the data prior to learning the subword model. We tested vocabulary sizes of 8000 and 32000, as well as using shared or split vocabularies for the source and target languages. Character coverage was set to 0.9995, the recommended value for languages with extensive character sets such as Chinese and Japanese.

For the English → Japanese, Korean → Japanese, and Chinese → Japanese language pairs, we supplemented the corpora with back translation (from Japanese into each language), which is a common data augmentation technique in NMT (Sennrich et al., 2016). The back translations were produced by the NMT systems trained for the other three directions (Japanese → English, Korean, and Chinese).

2.3 Models

Our NMT systems were standard base Transformer models trained using the Marian NMT framework (Junczys-Dowmunt et al., 2018). We trained separate, unidirectional models for each language pair. Hyperparameters such as label smoothing, dropout, learning rate, batch size, number of encoder/decoder layers, number of attention heads, embedding dimensionality, etc., were held fixed across all language pairs. The validation frequency was every 500 updates, and training was continued for 50 epochs or until the primary validation metric (ce-mean-words, or mean word cross-entropy score) failed to improve for five consecutive checkpoints. Our models were trained on AWS P3 instances using 4 NVIDIA Tesla V100 GPUs.

3 Results

Our results show that for most language pairs, a shared vocabulary of size 8,000 achieved the best performance. For the Korean → Japanese and Japanese → Korean language pairs, using a vocabulary size of 32,000 produced better results. Using a split vocabulary for these language pairs also resulted in better performance, whereas a shared vocabulary was advantageous for all other language pairs. In all cases, the inclusion of back translated training data resulted in higher validation scores. Table 1 shows our results in terms of BLEU scores (Papineni et al., 2002) as calculated on our local machines. Due to differences in processing, these scores do not match the scores reported by the Organizers.

4 Discussion

In this shared task, we set out to identify a single configuration of hyperparameters that provided the best overall performance across all six language pairs. While this approach precluded the possibility of obtaining optimal performance for all language pairs, it afforded the opportunity to investigate which hyperparameters have similar effects on different language pairs, and which have varied effects on different language pairs. As different language pairs require different hyperparameters, any parameter that can be held fixed during the experimentation stage can create significant savings for companies training their own machine translation models.

For instance, variation in parameters such as learning rate, dropout, embedding dimensions, and tying the weights of the source and target embedding layers seemed to have similar effects on performance across all language pairs that we tested. Using back translated data to augment the training sets also appeared to be universally beneficial. However, the size of the vocabulary seemed to have quite different effects in different language pairs. We are not aware of any theoretical framework for explaining how the various

Language Pair	Split 32K	Split 32K + BT	Shared 32K	Shared 8K	Shared 8K + BT
EN → JA	23.2	26.6	23.8	23.8	**27.1**
JA → EN	38.9	-	39.4	**40.2**	-
KO → JA	46.6	**46.8**	46.7	45.6	45.6
JA → KO	**52.0**	-	50.8	-	-
ZH → JA	30.6	31.6	31.8	31.9	**32.9**
JA → ZH	46.2	-	37.6	**47.5**	-

Table 1: BLEU scores for different language pairs and different vocabulary configurations

hyperparameters interact to produce such different results, nor do we know of any way of predicting the optimal hyperparameters for a given language pair other than iterative experimentation.

If additional resources are used, several additional steps have also been shown to be effective at boosting performance, but were not employed in these experiments in order to maintain maximum simplicity. These additional steps include using an ensemble of models for decoding, using larger model sizes, performing word segmentation prior to creating the vocabularies, ordering the training data using the output of a language model (a technique referred to as curriculum learning), and employing an additional model for right-to-left re-ranking.

With minimal manual intervention, our models achieved results ranging from fair to excellent. The large variance in the relative performance of these systems shows that no "one-size-fits-all" yet exists for the problem of machine translation. Despite monumental advances in the field over the past several years, achieving optimal performance requires careful selection of hyperparameters, and different configurations are required for different languages.

References

Kudo, Taku, and John Richardson. 2018 "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing." In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++ http://www.aclweb.org/anthology/P18-4020.

Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing 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.

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*, pages 5998–6008.