Kakao Enterprise's WMT21 Machine Translation using Terminologies Task Submission

Yunju Bak[1] **Jimin Sun**[2*] **Jay Kim**[1] **Sungwon Lyu**[1] **Changmin Lee**[1]
[1]Kakao Enterprise [2]Carnegie Mellon University
[1]{juliette.y,jay.ka387,james.ryu,louis.cm}@kakaoenterprise.com
[2]jimins2@cs.cmu.edu

Abstract

This paper describes Kakao Enterprise's submission to the WMT21 shared Machine Translation using Terminologies task. We integrate terminology constraints by pre-training with target lemma annotations and fine-tuning with exact target annotations utilizing the given terminology dataset. This approach yields a model that achieves outstanding results in terms of both translation quality and term consistency, ranking first based on COMET in the En→Fr language direction. Furthermore, we explore various methods such as back-translation, explicitly training terminologies as additional parallel data, and in-domain data selection.

1 Introduction

We participate in the WMT21 Machine Translation using Terminologies Task in four language directions, English→French (En→Fr), English→Chinese (En→Zh), English→Korean (En→Ko) and Czech→German (Cs→De).

1.1 Task description

The recent COVID-19 pandemic has raised the urgency to translate and distribute the latest medical information worldwide. However, despite recent advances in neural machine translation (NMT), translation in such emerging domains remains a challenge, as it is unaffordable to collect fair amounts of quality in-domain parallel data in a short time. As an alternative, word- or phrase-level dictionaries of key terms are relatively easier to obtain. These dictionaries are prevalent in commercial settings, where customers specify domain-specific jargon that human translators can attend to. However, incorporating pre-specified dictionaries effectively into NMT models is a non-trivial problem, as NMT is inherently trained without explicit constraints compared to statistical approaches.

In this context, the shared task of Machine Translation using Terminologies is held in five language directions at WMT21. The task assumes a realistic scenario where parallel and monolingual data are abundant in generic domains (e.g., news, web crawl), but only hundreds of word- or phrase-level term dictionaries are available in the domain of interest — COVID-19. Technically, this poses a challenge as we must impose terminology constraints without hurting general translation quality, while only 1.5% of parallel data contain the provided terminologies. Additional issues such as the $1:N$ mapping of term translations further complicate the problem.

Evaluating MT systems in specialized domains diverge from general MT evaluation in that overall translation quality may not ensure the translation accuracy of domain-specific terms. This potential gap calls the need for evaluation metrics that directly assess the consistent use of terms. Concretely, three metrics proposed in Alam et al. (2021) are employed in this task – Exact-Match Accuracy, Window Overlap, and Terminology-biased Translation Edit Rate (TERm). The suggested metrics complement general translation accuracy measured by standard MT metrics (BLEU, chrF, BERTscore, COMET) by validating whether terms are translated faithfully according to the dictionary.

Specifically, human-labeled COVID-19 related term dictionaries are released in four language directions (En→Fr, En→Zh, En→Ko, En→Ru), with around 600 terms for each direction. Exceptionally, the dictionary for Cs→De is constructed automatically and consists of 5,601 parallel terms.

1.2 Related work

Word- or phrase-level constraints have often been introduced to NMT via constrained decoding to reinforce specific tokens in the output sequence.

*Work done during the author's internship at Kakao Enterprise.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 804–812
November 10–11, 2021.

Combined with terminology dictionaries, constrained decoding integrates the target side terms as decoding-time constraints (Hokamp and Liu, 2017; Anderson et al., 2017; Post and Vilar, 2018).

Subsequent work has shown that adding inline annotations to the source sentence as soft constraints can improve performance and time complexity when employed with additional source factor streams (Dinu et al., 2019; Bergmanis and Pinnis, 2021). Similarly, a merging approach by adding markers without modifying the model has also proven to be effective (Wang et al., 2019).

2 Data

2.1 Cleaning

Both monolingual and parallel corpora of all languages are preprocessed according to the following pipeline. First, we remove non-utf8 or non-printable characters. Second, we unescape HTML characters such as `>`. Finally, we normalize variations in spaces and punctuation marks. All cleaning steps are done with Moses scripts (Koehn et al., 2007). We also use the Moses tokenizer, but only for European languages (En, Fr, Cs, De) since Asian languages (Zh, Ko) require language-specific tokenizers that consider the characteristics of each language.

2.2 Filtering

Web-crawled data are notorious for being noisy. To prevent defective data from undermining performance, we filter both parallel and monolingual data with diverse methods.

Bi-text We filter the provided parallel data with several heuristics. We first eliminate pairs that contain empty lines or identical content in both source and the target side. We filter pairs that contain overly long sentences (250 words) or excessively long words (50 characters). The pairs that have a word count ratio larger than four are also omitted. We refer to previous literature to set statistical thresholds of each rule. Lastly, we only use pairs of which both sides are identified as the correct language with a language identification tool. Specifically, we use fastText (Joulin et al., 2016, 2017).

In addition, for En→Ko, we filter out mislabeled bi-text which we found manually, that seemed as byproducts of web-crawl in the source or target side. For instance, the pattern "*YYYY년 MM월 DD일에 확인함*", which means "*Confirmed in YYYY/MM/DD*", was found instead of the correct labels in 20,909 samples. The final dataset sizes are shown in Table 1.

	En-Fr	En-Zh	En-Ko	Cs-De
Parallel	158M	62M	13M	15M
+ Filter	149M	-	12M	13M

Table 1: Dataset sizes of parallel corpora before and after filtering in each language pair. For En-Zh, we did not apply rule-based filtering.

Mono-text We used monolingual text for two language pairs (En→Ko, En→Fr) to augment existing parallel corpora via back-translation (Sennrich et al., 2016a). The back-translation procedure is described in Section 3.2.

For En→Ko, we do not apply any filtering schemes as the size of the Korean monolingual corpus is small (14M sentences).

On the contrary, for En→Fr, using the entire French monolingual corpora (8.5B) for back-translation is unwieldy, considering the time and computation required to infer all samples. Hence, we filter the corpus and select in-domain, COVID-19 related data to maintain a reasonable size for inference and training.

We filter French monolingual data in three steps. First, we roughly filter the data with rule-based methods that are similar to those of bi-text filtering. Second, we choose sentences that contain terms in the terminology dictionary (8.5B → 725M). Lastly, we use the Moore and Lewis (Moore and Lewis, 2010) method to find samples that are more similar to the term-related samples. Specifically, we train an in-domain language model with sentences that contain terminologies from the En-Fr parallel corpus. A general-domain language model is also trained with samples chosen randomly from the En-Fr parallel corpus. For both models, we use KenLM (Heafield, 2011) to train 5-gram language models with modified Kneser-Ney smoothing. Finally, top-k sentences with the highest scores are chosen (725M → 160M).

3 Approaches

3.1 Baseline

We explore two baseline approaches that differ by their training data. First, models are trained with

solely the parallel data described in 2.2. This baseline does not utilize the terminology dictionary.

Second, we take a naïve approach to leverage the term dictionary – including the provided terms as additional parallel data to train the model. For $1:N$ mappings of term translations, we flatten them into N distinct pairs. We refer to this approach as the "explicit" model in the following sections as we "explicitly" augment the training dataset with terminology dictionaries.

3.2 Back-translation

We incorporate back-translated monolingual data for two language directions: En→Fr and En→Ko.[1] We train reverse translation models (Fr→En, Ko→En) with the same parallel corpora and training configuration used to train our baseline models covered in Section 4.2. Back-translated samples are inferred with beam search of beam size 4, and a length penalty of 0.6.

For En→Fr, we use back-translated corpora for Exact Target Annotation fine-tune. We revisit the details of this procedure in Section 3.4.

For En→Ko, we train the back-translation model from scratch using both parallel and back-translated text. During training, we upsample the parallel corpus twice as frequently as the back-translated text.

3.3 Target Lemma Annotation

To integrate terminology constraints, we employ Target Lemma Annotation (**TLA**) of Bergmanis and Pinnis (2021), which helps the model learn how to copy-and-inflect inline annotations. At training time, we randomly select target lemmas and inject them into the source sentence behind the corresponding source word(s).

Specifically, we adopt a simple approach where we modify the input data but not the model. This differs from the method described in Bergmanis and Pinnis (2021), which uses additional input streams to denote the annotated tokens. In detail, we introduce three special tokens **<b>**, **<t>**, and **</t>** which respectively indicate the start of annotated source tokens, the start of target lemma tokens and the end of target lemma tokens. An example is shown in Table 2.

Following the training data annotation procedure of Bergmanis and Pinnis (2021), we first lemmatize and mark part-of-speech tags of the target sentences, using spaCy (Honnibal et al., 2020) instead of the pre-trained Stanza model (Qi et al., 2020) due to the time complexity. We then obtain word alignments using fast_align (Dyer et al., 2013) and randomly annotate verbs or nouns with their corresponding target lemma. To set annotation thresholds, we refer to Bergmanis and Pinnis (2021) – [0.6, 1.0) for sentence-level and [0.0, 1.0) for word-level. The annotated and original data are fed into the model with a proportion of 1:1.

At test time, we provide soft terminology constraints by annotating source terms with their corresponding target terms retrieved from the terminology dataset. Terminology entries are identified with the longest word-sequence match in the source sentence. If there exist several target terms for one source term, we randomly select one candidate.

Original Source	EN	and are you having any of the following **symptoms** with your chest pain?
Annotated Source	EN	and are you having any of the following **<b> symptoms <t> symptômes </t>** with your chest pain?
Target	FR	et avez-vous l'un de **symptômes** suivants en plus de vos douleurs thoraciques ?

Table 2: An example of using special tokens for inline annotations. Inline annotations are marked in bold. **<b>**, **<t>**, **</t>** denote the start of the annotated source tokens, the start of the target lemma tokens, and the end of the target lemma tokens.

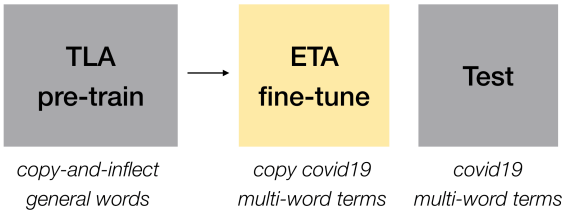

Figure 1: The steps in our TLA pre-train → ETA fine-tune approach and the objective of each phase.

3.4 Exact Target Annotation (Fine-tune)

We adopt Exact Target Annotation (ETA) designed by Dinu et al. (2019) to fine-tune the TLA model pre-trained as in Section 3.3. ETA injects the exact target-side translation of a terminology entry into the source sentence using inline annotations. Note that we utilized the whole terminology dataset during training, unlike Dinu et al. (2019), since the task allows the use of the terminology dataset at

[1] We do not incorporate back-translated corpora of Cs-De and En-Zh due to time constraints.

training time.

While TLA learns to copy-and-inflect general words, our terminology dataset is domain-specific. We aim to fill the domain gap by constructing fine-tuning data in which terminology entries are present on both the source and target sides. As a result, 750K samples from the parallel data and 10M samples from the back-translated data are selected. We upsample the parallel corpus by eight times.

Another discrepancy between training and test time annotation in TLA is that TLA engages a single target word to the corresponding source word(s), whereas many of the actual terms are multi-word expressions in both source and target sides. We expect ETA fine-tune to alleviate the problem since ETA annotates target terms in verbatim. The pretrain-finetune phases are outlined with their motivation in Figure 1.

Specifically, we follow the annotation strategy of Dinu et al. (2019), where we annotate only when both the source side term t_s and the target side term t_t are present. When a sentence contains multiple matches overlapping each other, we keep the longest match.

The difference between Dinu et al. (2019) and our method is that we annotate with three special tokens as described in Section 3.3. Instead of randomly deciding whether to annotate or not, we annotate all matches. We then combine the annotated data with its original data and use it for training with a proportion of 1:1. The annotation procedure at test time is also equivalent to Section 3.3.

4 Experiments

4.1 Evaluation setting

Evaluation of the models is done using the evaluation script [2] and the development dataset, both provided by the task organizers. We select the best models by considering all metrics provided by the evaluation script.

For evaluation, we tokenize our outputs so that they resemble the tokenization setup of the development dataset. For En→Fr and Cs→De, we use the Moses toolkit (Koehn et al., 2007). For En→Zh, we apply the Jieba tokenizer.[3]

Before submitting the test set translations, we handle rare target-side tokens decoded as <unk> by simple substitutions, which we found to work well during evaluation even without incorporating external methods such as word alignments. When the number of <unk> tokens are equal on both sides, we copy the original source-side tokens to the target slots in the same order. After replacing rare tokens, outputs are detokenized using the Moses toolkit (Koehn et al., 2007).

4.2 Experimental details

For En→Fr and Cs→De, we pre-tokenize the data using the Moses toolkit (Koehn et al., 2007). We use sentencepiece (Kudo and Richardson, 2018) to learn a joint byte pair encoding (BPE) with vocabulary size 40K (En→Fr) and 32K (Cs→De). For En→Ko, We pre-tokenize Korean sentences with Mecab (Kudo, 2005) without space tokens as suggested in Park et al. (2021) and use sentencepiece to learn a BPE model with vocabulary size 32K for each language side. For En→Zh, we first convert characters possibly in traditional Chinese to simplified Chinese text using hanziconv [4] and. Then, we pre-tokenize the data using the Jieba tokenizer [3]. We then use subword-nmt (Sennrich et al., 2016b) to train BPE on combined Chinese and English corpus and build separated vocabularies. The final vocabulary size is 44K for Chinese and 32K for English.

For all language directions, we employ the Transformer architecture (Vaswani et al., 2017) implemented in fairseq (Ott et al., 2019). The specific training and generation configurations can be found in Appendix A.

Since TLA relies on the word-aligner's performance, we did not apply TLA pre-training and ETA fine-tuning for En→Ko and En→Zh. Given that both are linguistically distant language pairs, we assumed that the word-aligner's performance would not be sufficient enough to guarantee improvements from TLA.

We start ETA fine-tuning from the TLA checkpoint saved at 750,000 steps for En→Fr and 200,000 steps for Cs→De, chosen based on BLEU scores and Exact Match Accuracy. To evaluate the TLA and ETA fine-tuned models, we run annotation using the terminology tags provided with the development dataset, which is different from the test annotations described in 3.3.

For En→Ko and Cs→De, we use an ensemble of models that utilize back-translation, explicit training, and data augmentation. The exact ensemble

[2] https://github.com/mahfuzibnalam/terminology_evaluation

[3] https://github.com/fxsjy/jieba

[4] https://github.com/berniey/hanziconv

System	BLEU	Exact Match	Window Overlap (Window 2/3)	1-TERm
En-Fr				
Baseline	47.83	0.882	**0.31/0.301**	0.628
TLA	47.07	0.915	0.282/0.275	0.611
TLA w/o annotation	47.84	0.881	0.305/0.297	0.617
TLA + ETA fine-tune (bi-text only)	47.47	**0.932**	0.298/0.289	0.615
TLA + ETA fine-tune	**48.16**	0.929	0.307/0.30	**0.631**
En-Zh				
Baseline	29.08	0.803	**0.192**/0.194	0.418
Explicit	**29.81**	**0.805**	**0.192/0.197**	**0.431**
En-Ko				
Baseline	12.04	0.412	0.038/0.037	0.129
Baseline + BT	14.14	0.417	0.039/0.037	0.172
Explicit	12.27	0.42	0.034/0.032	0.151
Explicit + BT	14.24	**0.464**	0.04/0.04	**0.184**
Ensemble	**14.56**	0.454	**0.043/0.042**	0.178
Cs-De				
Baseline	30.95	0.832	0.41/0.398	0.434
Explicit	30.77	0.833	0.408/0.396	0.433
Ensemble	**32.47**	0.848	**0.429/0.416**	**0.445**
TLA	28.46	**0.924**	0.281/0.272	0.395
TLA + ETA fine-tune (bi-text only)	30.14	0.889	0.353/0.342	0.417

Table 3: Evaluation results for each task language pair. Highest scores are **boldfaced**. Rows in gray indicate our submitted systems for test evaluation.

configurations are detailed in Appendix B.

5 Results

Table 3 reports the evaluation results of the four language pairs that we participated in.

5.1 English→French

The TLA model improves Exact Match Accuracy but shows deteriorated performance on all other metrics compared to the baseline. Notably, the degradation stems from the test-annotation method – test scores are comparable to the baseline when tested with raw text (without test-annotation) on the same TLA model.

On the other hand, under the same test-annotation condition, the ETA fine-tuned model recovers the performance loss and even boosts the BLEU score, Exact Match Accuracy, and the 1-TERm score compared to both the baseline and the TLA model. TLA + ETA fine-tune outperforms the baseline by 0.33 points, 4.65%, and 0.24% on BLEU, Exact Match, and 1-TERm, respectively.

In addition, we run a simple ablation experiment by using only bi-text data during ETA fine-tuning: TLA + ETA fine-tune (bi-text only). The results are indistinguishable from the original TLA + ETA fine-tune, which is fine-tuned with data from both bi-text and mono-text. This result supports that the performance gain stems not only from the use of monolingual data, which was unseen during TLA pre-training.

5.2 English→Chinese

We compare two approaches – baseline and explicit, and observe that adding the term pairs explicitly to training improves both general translation performance (+0.73 BLEU) and term consistency (+2.29% 1-TERm) compared to the baseline.

5.3 English→Korean

Back-translation yields performance gains across all metrics with considerable improvements, particularly in BLEU and 1-TERm. The explicit model also brings modest improvements to Exact Match

Language	COMET			Exact Match Accuracy			Number of
	Ours	Best	Rank	Ours	Best	Rank	Submissions
En-Fr	**0.781**	-	1	0.95	0.974	4-6	22
En-Zh	0.229	0.716	8	0.645	0.886	7-8	8
En-Ko	**0.581**	-	1	**0.569**	-	1	1
Cs-De	**0.694**	-	1	0.866	0.871	1-2	2

Table 4: Official task results of our submitted systems. Scores, where our system ranked 1st, are bold-faced. In other cases, the best scores from other submissions are shown for comparison.

Accuracy and 1-TERm. Finally, our ensemble model that combines these approaches demonstrates the best performance across all metrics, raising the BLEU score by 2.52 points, Exact Match Accuracy by 4.2%, Window Overlap by 0.43% and 0.54% for windows 2 and 3 respectively, and 1-TERm by 4.88 points.

5.4 Czech→German

We discover that the explicit model does not bring significant gains compared to the baseline model. This trend contradicts other language directions, where we observed at least modest improvements over their respective baselines. We suspect the differences lie in how the terminologies are generated; Cs→De terminologies are constructed automatically, whereas, for other language directions, the terminologies were annotated manually.

Our ensemble model improves upon the baseline model by 1.5 BLEU points, 1.6% Exact Match Accuracy, 1.84% and 1.74% Window Overlap for window sizes 2 and 3, and 1.1 points in 1-TERm.

We also attempted to apply TLA pre-training + ETA fine-tuning to Cs→De as done in En→Fr. In our preliminary experiments, while some metrics improved, we observed Exact Match Accuracy deteriorate after 1,000 steps of TLA training, unlike En→Fr, possibly due to the automatic creation pipeline of Cs→De terminologies. Therefore, we did not further explore this direction during our task participation. However, subsequent experiments after the deadline revealed that TLA, when followed by ETA fine-tuning, has its advantages in finding a balance between BLEU and Exact Match Accuracy, supporting our findings in En→Fr.

5.5 Official task results

We present our official submission results in Table 4. Despite the trade-off between general translation quality (COMET) and term consistency (Exact Match Accuracy), our approach strikes at the right balance between the two criteria for En→Fr. Out of 22 submissions in this direction, our system ranks 1st in COMET. According to Exact Match Accuracy, our system performs roughly comparable to the best system, ranking 4-6th. For En→Zh, our system ranks 8th in both metrics out of 8 submissions. For En→Ko, our submission is the only submission. For Cs→De, our submission ranks 1st in terms of COMET and 1st-2nd for Exact Match Accuracy out of 2 submissions.

6 Conclusion

We participate in four language directions for the shared task WMT21 Machine Translation Terminologies. To this end, we explore various techniques, including back-translation, explicitly training with term pairs along with other parallel data, and in-domain data selection to improve translation performance in the COVID-19 domain.

In particular, for En→Fr and Cs→De, we find that TLA outperforms the baseline in terms of Exact Match Accuracy by leveraging terminology constraints. However, all other metric scores (BLEU, 1-TERm) plummeted, implying that the overall translation quality was compromised. We recover this performance loss by introducing a new technique – fine-tuning with ETA, and achieve significant improvements in both general translation quality and terminology consistency. We leave it to future work to validate our approach in other languages and reveal the factors behind the benefits of ETA fine-tuning precisely, hopefully, to discover a more suitable design to impose terminology constraints.

References

Md Mahfuz Ibn Alam, Antonios Anastasopoulos, Laurent Besacier, James Cross, Matthias Gallé, Philipp

Koehn, and Vassilina Nikoulina. 2021. On the evaluation of machine translation for terminology consistency. *CoRR*, abs/2106.11891.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Taku Kudo 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*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Takumitsu Kudo. 2005. Mecab : Yet another part-of-speech and morphological analyzer.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

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*.

Sungjoon Park, Jihyung Moon, Sung-Dong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Tae Hwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Young kuk Jeong, Inkwon Lee, Sang gyu Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Lucy Park, Alice H. Oh, Jung-Woo Ha, and Kyunghyun Cho. 2021. Klue: Korean language understanding evaluation. *ArXiv*, abs/2105.09680.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Tao Wang, Shaohui Kuang, Deyi Xiong, and António Branco. 2019. Merging external bilingual pairs into neural machine translation.

A Training configuration

```
fairseq-train
  task : translation
  arch : transformer_wmt_en_de_big
  lr : 0.0005
  lr-scheduler : inverse_sqrt
  warmup-updates : 4000
  warmup-init-lr : 1e-07
  optimizer : adam
  adam-betas : (0.9, 0.98)
  update-freq : 8
  dropout : 0.1
  weight-decay : 0
  criterion : label_smoothed_cross_entropy
  label-smoothing : 0.1
  fp16 : True

fairseq-train (ETA fine-tune)
  lr : 1e-06
  lr-scheduler : fixed
  warmup-updates : 0

fairseq-generate
  beam : 4
  lenpen : 0.6
```

B Ensemble Configuration

For En→Ko, we use an ensemble of four models trained with different configurations:

- Baseline + Back-translation
- Baseline + Back-translation + Rule-based filtering
- Baseline + Back-translation + Explicit
- Baseline + Back-translation + Explicit (Parallel corpus upsampling with ratio 2)

For Cs→De, we use an ensemble of four models trained with different configurations. The third model concatenates the previous and next sentence for additional context with probability of 0.1:

- Baseline
- Baseline + Rule-based filtering
- Baseline + Two sentences concatenation (0.1)
- Baseline + Explicit