VARCO-MT: NCSOFT's WMT'23 Terminology Shared Task Submission

Geon Woo Park, Junghwa Lee, Meiying Ren, Allison Shindell, Yeonsoo Lee
NCSOFT NLP Center
{parkku01, jleehhh0217, mia1211, shindell, yeonsoo}@ncsoft.com

Abstract

A lack of consistency in terminology translation undermines quality of translation from even the best performing neural machine translation (NMT) models, especially in narrow domains like literature, medicine, and video game jargon. Dictionaries containing terminologies and their translations are often used to improve consistency but are difficult to construct and incorporate. We accompany our submissions to the WMT '23 Terminology Shared Task with a description of our experimental setup and procedure where we propose a framework of terminology-aware machine translation. Our framework comprises of an automatic terminology extraction process that constructs machine translation datasets with terminology dictionaries in low-supervision settings and two model architectures with terminology constraints. Our models outperform baseline models by 21.51%p and 19.36%p in terminology recall respectively on the Chinese to English WMT'23 Terminology Shared Task test data.

1 Introduction

The WMT'23 Terminology Shared Task aims to assess machine translation models' abilities to leverage additional information. A terminology dictionary is provided with each line of source text. This is particularly useful for terminology consistency. The WMT'23 Terminology Shared task includes the language pairs Chinese-English, English-Czech, and German-English. We focus our submission on the Chinese-English pair. The task consists of three modes, as shown in Table 1.

Mode 1 assesses translation quality without additional terminology information. Mode 2 assesses translation quality with additional terminology information. Mode 3 assesses translation quality with a glossary containing random non-terminology.

In this paper, we describe our model building process for terminology translation from data preprocessing to model evaluation. We present two Transformer-based encoder-decoder models: Terminology Self-selection Neural Machine Translation (TSSNMT) and ForceGen Transformer (ForceGen-T). TSSNMT uses a shared encoder with a gating mechanism (Bapna and Firat, 2019), allowing the model to determine the weights of the source sentence and terminologies to use during generation. ForceGen-T enforces a decoder to generate the terminologies via force decoding (Reheman et al., 2023) and copy mechanism (Song et al., 2019), which enables the model to attend to terminologies during generation. Both models significantly outperform the baseline model.

2 Related works

Previous work on enhancing machine translation with pre-defined terminology encompasses three primary approaches.

First, a data-driven approach where terminologies are appended to input sentences (Dinu et al., 2019; Song et al., 2019). Song et al. (2019) suggest using copy mechanism to instruct the model to replicate the target terminology during the generation process.

Second, an alternative approach focuses on manipulating the model architecture. Bapna and Firat (2019) have used input sentences and their corresponding retrieved translation pairs to encode conditional source target memory. This approach uses a gated multi-source attention mechanism, which takes the encoded representation and the hidden state of the source as input, thereby steering the model toward the generation of the intended translation.

Third, efforts have been directed at tailoring the decoding process to incorporate terminologies. Hokamp and Liu (2017) and Post and Vilar (2018) have introduced constrained decoding techniques that reinforce the translated output's pre-specified terminologies.

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 919–925
December 6–7, 2023. ©2023 Association for Computational Linguistics

Mode	Source Input	Glossary Input	Target Output
1	脱离这些影响的建筑,17世纪的建筑师伊尼戈琼斯和克里斯托弗·雷恩牢固确立了在英国的古典主义.	-	Architects Inigo Jones and Christopher Ren strongly established classicalism in England in the 17th century, free from these influences.
2		{“en”: “Christopher Wren”, “zh”: “克里斯托弗·雷恩”}	Inigo Jones and **Christopher Wren**, two architects from the 17th century, strongly established classical architecture in England, free from these influences.
3		[“en”: “firmly”, “zh”: “牢固”, “en”: “free”, “zh”: “脱离”]	Building designs that were **free** from these influences, 17th century architects Inigo Jones and Christopher Ren **firmly** established classical architecture in England.

Table 1: Different Mode Scenarios

3 Data Process

The WMT’23 Terminology Shared Task is a constrained track, following the same rules of data usage as the WMT’23 General MT Task, forbidding the use of external data. However, unlike the WMT’21 Terminology Task, the provided training data lacks terminology information, and need to be artificially constructed.

3.1 Data Filtering

We first filter noisy data. We referenced the data cleaning methods described in the WMT’21 Terminology submissions (Molchanov et al., 2021; Wang et al., 2021).

1. Remove pairs that contain sentences that:
   (a) are empty, too short, too long.
   (b) are contain only symbols.
   (c) are at least 3 times longer than their counterpart.
2. Delete text pairs identified to be the wrong language. We used a combination of our in-house language detector for short texts and LangID (Lui and Baldwin, 2012) for long texts.
3. Remove pairs outside of a selected cosine similarity scope of latent vectors constructed by the LaBSE model (Feng et al., 2022).

See Appendix A for the amount of data filtered and the resulting performance comparison.

3.2 Word Alignment

After filtering data, we tokenize and word-align the text to extract desired terminology pairs for Modes 2 and 3. The overall process is described in Figure 1. We use our in-house tokenizer, referring to the tagging schema from Luo et al. (2019) for Chinese and the Moses (Koehn et al., 2007) tokenizer for English. Next, the tokenized parallel data is fed into a LaBSE (Feng et al., 2022) based word aligner, AccAlign (Wang et al., 2022). AccAlign generates pairs of indices for words from the source and target text, which are then utilized in extracting terminology pairs.

3.3 Terminology Extraction: Mode 2

The terminology extracted for Mode 2 are named entities, excluding time and number expressions. We use SpaCy’s (Honnibal and Montani, 2017) zh_core_web_lg as the Chinese NER model and en_core_web_md as the English NER model. Furthermore, we consider Chinese four-character idioms extracted by our in-house tokenizer. The idioms are added as additional Mode 2 candidates. Next, we reference our word alignment results from Section 3.2 to map candidates to their corresponding targets.

AccAlign occasionally fails to align multi-word terminology completely, which poses an issue for Chinese idioms. To account for this, we implement a soft matching strategy to interpret AccAlign’s output indices, where we extract the entire phrase if the aligner maps the beginning and end indices, even when alignment is not complete in the middle of the phrase. We use strict matching for named entities, which only extracts words that appear in the alignment results.

To guarantee that our extracted terminology is accurate and exhaustive, we repurpose the provided training data source WikiTitles as an additional resource for terminology pairs. For each pair in WikiTitles, we check whether a term and its translation were present on both sides of the parallel text and add the relevant term pairs into the Mode 2 glossary.

3.4 Terminology Extraction: Mode 3

The terminology extracted for Mode 3 is intended to be relatively random yet accurate pairs from the parallel text. For simplicity, we exclusively

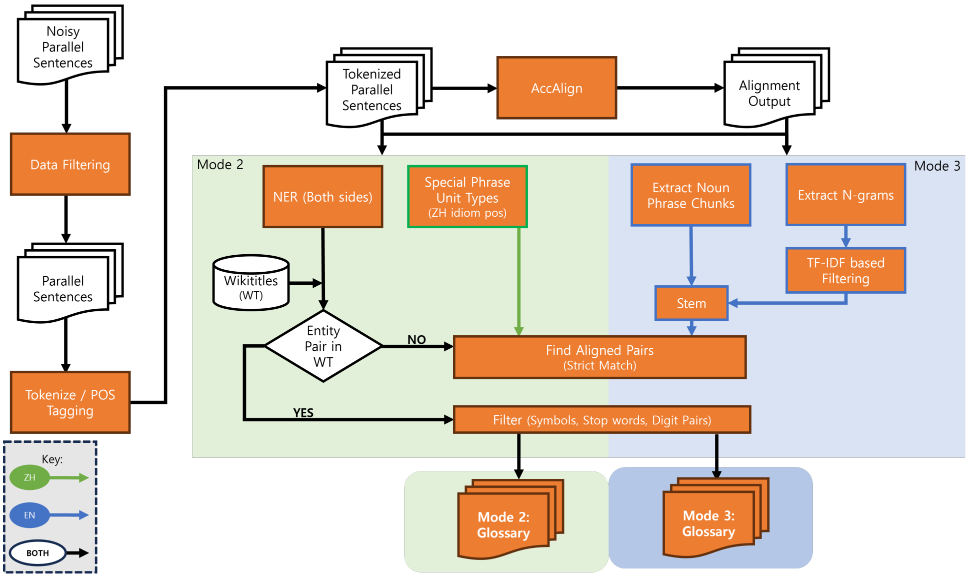

Figure 1: Data Construction Process

processed from the English side when generating Mode 3 candidates. We extracted n-grams ($1 \leq n \leq 4$) with high TF-IDF scores (Sparck Jones, 1988), as well as noun phrases (Loria, 2018), as the Mode 3 candidates. We take all Mode 3 candidate pairs that contain an appropriately aligned terminology, stem the English terms using `nltk` (Bird et al., 2009), and then randomly select of a maximum of ten term pairs for the Mode 3 glossary.

3.5 Development Data

The official Chinese-English development data is relatively small, thus we supplement it with a subset of the allowed data. We construct a supplemental development data with a random proportional sample from each provided training data source, which consists of 1,000 identical sentences throughout the three modes. Furthermore, we construct terminology for the different modes according to the above mentioned process. Additionally, we filter stop words and terms in neither the source nor target texts.

4 Models

This section presents two distinct models designed to incorporate terminologies into NMT models. The first model, Terminology Self-selection Neural Machine Translation (TSSNMT), employs a shared encoder architecture featuring a gating mechanism. This mechanism empowers the model to make decisions regarding the proportion of the source sentence and the terminologies to be processed during generation. The second model, ForceGen Transformer (ForceGen-T), takes a more straightforward approach, utilizing a standard Transformer model with force decoding (Reheman et al., 2023) and copy mechanism (Song et al., 2019). This approach enforces the model to initially generate the predefined terminologies before generating the remainder of the sentence. Copy mechanism is applied to replicate source-side target terminologies in the output.

4.1 TSSNMT

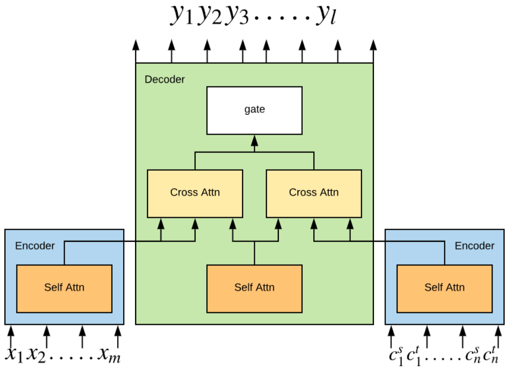
$y_1 y_2 y_3 \ldots\ldots y_l$
Decoder
gate
Cross Attn
Cross Attn
Encoder
Self Attn
Self Attn
Encoder
Self Attn
$x_1 x_2 \ldots\ldots x_m$
$c^s_1 c^t_1 \ldots\ldots c^s_n c^t_n$

Figure 2: TSSNMT Model Structure. x, y, and c denote source, target and corresponding terminology respectively.

We have implemented the TSSNMT model with minor changes to the transformer architecture. The model has two encoders, as shown in Figure 2. Each encoder receives input in source sentences and source-target pair terminologies. These two en-

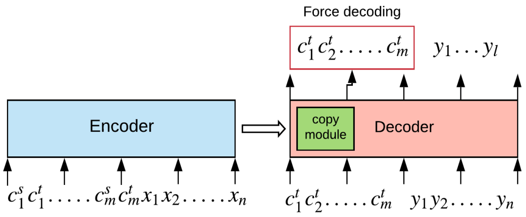
Force decoding
$c^t_1 c^t_2 \ldots\ldots c^t_m$
$y_1 \ldots y_l$
Encoder
copy module
Decoder
$c^s_1 c^t_1 \ldots\ldots c^s_m c^t_m x_1 x_2 \ldots\ldots x_n$
$c^t_1 c^t_2 \ldots\ldots c^t_m$ $y_1 y_2 \ldots\ldots y_n$

Figure 3: ForceGen model structure. x, y, and c denote source, target and corresponding terminology respectively.

coders share parameters and encode both the source sentences and the terminologies. The decoder calculates cross-attentions with these two encoder hidden states separately and then projects them to a gating mechanism (Bapna and Firat, 2019). Sharing parameters allows the model to decide weight distribution across the source and the terminology during generation.

4.2 ForceGen Transformer

We tailor a Transformer-based model to ensure the appearance of given terminologies in the generated output. Modifying the input format and decoding process incorporates copy mechanism on the source side, allowing it to copy the target terminology from the provided terminology pairs. Table 2 refers to the input sentence for this purpose.

During decoding, the model is reinforced to generate the given terminologies in a teacher-forcing manner. This approach aligns with findings by Reheman et al. (2023), who force a model to generate Translation Memory (TM) to enhance model performance. Instead, we provide the model with the terminologies, expecting it to consider them attentively during the decoding process. Once the terminologies are successfully generated, the model decodes the remainder of the input text. Copy mechanism enforces the source-side target constraints into the output. This approach is inline with that of Song et al. (2019), where copy mechanism significantly improves the ratio of terminology occurrences in the output.

We conduct preliminary experiments by training our models using the IWSLT17 (Cettolo et al., 2017) Chinese-English data and MUSE dictionary (Conneau et al., 2017) to assess the impact of the copy mechanism and force decoding. The primary objective is to determine whether the copy mechanism and the force decoding technique could complement each other. The outcomes, as presented in Table 3, reveal that the model yields the most favorable results when both the copy mechanism and force decoding are concurrently applied. This finding underscores the benefit of replicating source-side target terminology during the generation process, as it significantly aids in generating pre-specified terminologies during the force decoding phase. Consequently, when generating output after force decoding, the model effectively focuses on the target terminologies generated during decoding, facilitating successful incorporation of these terminologies into the final production. We apply both methods to our model, ForceGen-T.

5 Experiments

5.1 Evaluation setting

5.1.1 Pseudo test data

Both the provided WMT Terminology test data and blind data for the Chinese-English language pair contain only Chinese source lines and no target, so to evaluate the model, we constructed artificial target answers using ChatGPT (Ouyang et al., 2022) and reviewed the produced data manually. We then use this data as the test data to evaluate the model.

5.1.2 Evaluation metrics

The evaluation criteria for this translation task include overall terminology translation, terminology usage, and translation quality. We chose SacreBLEU (Post, 2018) , COMET (Rei et al., 2020), chrF (Popović, 2015) and Copy Success Rate (CSR). SacreBLEU and COMET scores are commonly used metrics in machine translation quality. For terminology translation and usage, we utilize CSR, which we define as the appearance rate of the desired terminology in the inferred text.

5.2 Experimental details

We use sentencepiece (Kudo and Richardson, 2018) to learn a joint byte pair encoding with a vocabulary size of 32K. Our preprocessing strategy involves pre-tokenizing Chinese data through an in-house Chinese tokenizer, while English data is exclusively tokenized using the Sentencepiece model. Please note that the training data tokenization process slightly differs from the data construction described in Section 3.

For all the experiments, we build upon the scale of the Transformer Big model (Vaswani et al.,

Source	郝仁, 人如其名, 是一个好人。
Term	{“en”: “Hao Ren”, “zh”: “郝仁” }
Modified source	郝仁<C> Hao Ren </C> 郝仁, 人如其名, 是一个好人
Modified target	Hao Ren </C> Hao Ren, as his name suggests, is a good man

Table 2: ForceGen training data sample.
<C>, </C> are the separation token that distinguishes the source sentence from the terminologies.

Model	COMET	SacreBLEU	chrF	CSR
Baseline	0.7274	18.78	42.09	78%
+Copy	0.7347	19.44	42.16	92%
+Force decoding	**0.7371**	**20.10**	**43.31**	**94%**

Table 3: Preliminary experiment results of ForceGen-T trained with IWSLT17 Chinese-English data.

Test data	Model	COMET	SacreBLEU	chrF	CSR
Test data	Baseline	0.6932	17.13	45.13	54.35%
	TSSNMT	0.7205	**23.04**	48.68	**75.86%**
	ForceGen-T	**0.7380**	22.02	**51.00**	73.71%
Blind data	Baseline	0.6918	16.55	45.88	65.07%
	TSSNMT	0.7181	**23.26**	49.18	83.45%
	ForceGen-T	**0.7336**	20.96	**51.57**	**89.38%**

Table 4: Experiment results. Please note that the scores are measured with Chat-GPT generated references.

2017) architecture implemented using our proprietary toolkit. This model consists of 12 encoder layers and 6 decoder layers, providing a strong foundation for effectively integrating specified terminologies into the output. The specific configuration of each approach varies according to the respective model specifications. We list detailed configurations in Appendix B.

6 Results

Table 4 shows the Chinese-English translation results on the WMT’23 Terminology Task. We compare two approaches - TSSNMT and ForceGen-T against the baseline Transformer Big model. Both TSSNMT and ForceGen-T significantly outperform the baseline model in all automatic evaluation metrics. Highly elevated CSR scores underscore the successful integration of provided terminologies into the translated output. In contrast, higher scores in various syntactic and semantic metrics (COMET, SacreBLEU, and chrF) indicate the fluency and adequacy of the generated translations. Within the test data, both TSSNMT and ForceGen-T exhibit similar performance levels. However, when evaluating based on the CSR score, TSSNMT surpasses ForceGen-T by approximately 2%p. In contrast, within the blind data, ForceGen-T consistently demonstrates superior scores compared to TSSNMT, with particularly notable advantages in CSR scores.

7 Conclusion

This paper presents the comprehensive procedure of our submissions for the WMT’23 Terminology Shared Task. Our approach involves meticulous refinement and pre-processing of the provided data, subsequently used to train our models. We investigate and implement two strategies for effectively integrating the given terminologies into the output, demonstrating their superior performance compared to the baseline. The result shows that our approach can significantly improve translation accuracy by increasing the recall of terminologies. As a future endeavor, we aim to extend the validation of our approach to other languages.

8 Limitations

In this paper, we propose two successful terminology integration approaches in NMT. We confirm that our models achieve significant performance gains over the baseline model. Still, it is essential to note that these observed improvements are specific to a particular language pair, Chinese to English. Therefore, further experiments on a wide range of language pairs, including those with morphologically complex structures, are needed to validate the broader efficacy of our approaches.

It is worth noting that the inference speed of ForceGen-T linearly correlates with the number of terminologies that need to be generated. ForceGen-T is forced to generate the given terminologies first in decoding, inevitably requiring additional inference time. Consequently, the inference speed of ForceGen-T is slower than that of the baseline Transformer model.

References

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. *arXiv preprint arXiv:1903.00058*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. *arXiv preprint arXiv:1906.01105*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Steven Loria. 2018. textblob documentation. *Release 0.15*, 2.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Zhiyuan Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455.

Alexander Molchanov, Vladislav Kovalenko, and Fedor Bykov. 2021. PROMT systems for WMT21 terminology translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 835–841, Online. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *arXiv preprint arXiv:1804.06609*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Abudurexiti Reheman, Tao Zhou, Yingfeng Luo, Di Yang, Tong Xiao, and Jingbo Zhu. 2023. Prompting neural machine translation with translation memories. *arXiv preprint arXiv:2301.05380*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing nmt with pre-specified translation. *arXiv preprint arXiv:1904.09107*.

Karen Sparck Jones. 1988. *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, page 132–142. Taylor Graham Publishing, GBR.

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.

Ke Wang, Shuqin Gu, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021. TermMind: Alibaba's WMT21 machine translation using terminologies task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 851–856, Online. Association for Computational Linguistics.

Weikang Wang, Guanhua Chen, Hanqing Wang, Yue Han, and Yun Chen. 2022. Multilingual sentence transformer as a multilingual word aligner. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2952–2963, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Appendix

Data	Num of raw data	Num of filtered data	ratio
Back-translated news	16,943,688	16,349,073	96.49%
CCMT Corpus(casia2015)	1,048,400	1,046,410	99.81%
CCMT Corpus(casict2011)	1,512,478	952,259	62.96%
CCMT Corpus(casict2015)	2,019,011	1,968,537	97.50%
CCMT Corpus(datum2017)	718,025	656,980	91.50%
CCMT Corpus(neu2017)	1,967,605	1,894,805	96.30%
News Commentary v18.1	311,904	309,410	99.20%
ParaCrawl v9	10,508,286	6,599,206	62.80%
UN Parallel Corpus v1.0	12,354,729	12,206,477	98.80%
WikiMatrix	2,276,736	1,035,916	45.50%
Total	**49,660,862**	**43,019,073**	**86.63%**

A.1: Data Filtering

Train data	COMET	sacreBLEU	chrF	CSR
raw data	0.5829	12.36	34.41	72.54%
filtered data	0.6445	15.04	40.58	73.64%

A.2: Comparison of WMT'23 raw data and filtered data on the test data

B Appendix

Training configuration	Hyper-parameters
embedding size	1024
num of encoder layers	12
num of decoder layers	6
num of heads	16
hidden size	1024
bottleneck size	4096
dropout rate	0.15
optimizer	fusedadam
learning rate	1.8
lr scheduler	noam
warm up step	4000
strategy	deepspeed_stage_2(Rajbhandari et al., 2020)

B.1: Training Configure