Combining Translation Memory with Neural Machine Translation

Akiko Eriguchi Microsoft
Spencer Rarrick Microsoft
Hitokazu Matsushita Microsoft
One Microsoft Way, Redmond, WA 98052 USA
{akikoe, spencer, himatsus}@microsoft.com

Abstract

In this paper, we report our submission systems (geoduck) to the Timely Disclosure task on the 6th Workshop on Asian Translation (WAT) (Nakazawa et al., 2019). Our system employs a combined approach of translation memory and Neural Machine Translation (NMT) models, where we can select final translation outputs from either a translation memory or an NMT system, when the similarity score of a test source sentence exceeds the predefined threshold. We observed that this combination approach significantly improves the translation performance on the Timely Disclosure corpus, as compared to a standalone NMT system. We also conducted source-based direct assessment on the final output, and we discuss the comparison between human references and each system's output.

1 Introduction

One of the desired features in automatic translation systems is the ability to flexibly make use of a translation memory to translate known sentences and phrase, while still allowing a more flexible Machine Translation (MT) model to translate less-familiar phrases and sentences without sacrificing quality. Koehn and Senellart (2010) explored methods of combining translation memories with statistical machine translation, and proposed a method to apply phrase fixing on the translation candidate retrieved from the translation memory. As for statistical machine translation models, Neural Machine Translation (NMT) models have been nowadays employed in large-scale production MT systems (Johnson et al., 2017; Hassan et al., 2018) due to its state-of-the-art performance in many languages.

There are several studies that combine translation memories with NMT models. Cao and Xiong (2018) introduced the idea of using a translation memory gating network with NMT models in a multi-encoder fashion so that the model can make full use of both training data and the retrieved data. Zhao et al. (2018) created phrase tables as recommendation memory and let the NMT models select the better translation. In Gu et al. (2018), the authors proposed a search-engine-guided NMT model, where a search engine first collects a small subset of relevant training translation pairs from the translation memory and NMT models are trained on the subset as well.

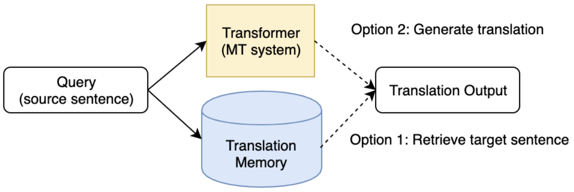


Figure 1: Overview of our proposed approach combining translation memory and NMT models.

In this paper, we combine a translation memory with an NMT model, simply choosing either the translation memory output or NMT output at inference time, depending on a similarity score of a given source sentence, and then investigate the effectiveness of this strategy. Figure 1 illustrates an overview of our proposed architecture. In Section 2, we first conduct an analysis of the Timely Disclosure task data set and report interesting characteristics. In Section 3, we explain our approach of combining a translation memory with an NMT model. We describe our experimental design in Section 4, and we report experimental results, human evaluation analyses, and discussion in Section 5. In Section 6, we conclude by summarizing our findings on the task and contributions of the paper.

Proceedings of the 6th Workshop on Asian Translation, pages 123–130
Hong Kong, China, November 4, 2019. ©2019 Association for Computational Linguistics

	size	# (Ja)	# (En)
Train.	1,403,995	583,805	709,358
Dev.	3,998	3,518	3,752
Devtest	4,014	3,753	3,483
Test	3,277	2,898	——

Table 1: Data statistics of Timely Disclosure Documents Corpus. # denotes the number of unique sentences in each language.

2 Analysis of Timely Disclosure Documents Corpus

We first analyzed the data set for the Timely Disclosure Task[1] to better understand trends in the data set. The corpus consists of past years' timely disclosure documents, and contains about 1.4 million Japanese-English sentence pairs. Table 1 reports the statistics of the full corpus. Here, we show the total size of parallel corpora, as well as the number of unique Japanese (Ja) and English (En) sentences in each data set. From this table, we can observe a large number of duplicated training examples on both the source and target sides. We also checked for duplicates between training data (Train.), development (Dev.) and devtest (Devtest), and found respectively 1,047 and 1,117 duplicated translation pairs between Train. and Dev. / Devtest data sets. Note that unique hash values are given to all translation pairs, which guarantees that these sentence-level translation pairs are independently sampled from the original documents. This finding is our motivation for combining translation memory retrieval with NMT models, and we investigate if this leads to an improvement in translation quality.

3 Combining Translation Memory with Neural Machine Translation

As observed in Section 2, approximately 26% of the development data set and 28% of the devtest data set are exact duplicates on both source and target of a sentence pair in the training data set. Considering this characteristic of the evaluation data sets, we use the entire training data set as a translation memory and allow the system to directly retrieve a translation candidate for each test sentence based on the best similarity score. If there is no translation candidates in the translation memory whose similarity score exceeds the threshold, we let the NMT models generate a translation and use it as a final output. We would like to mention that this kind of translation scenario is not specific to this task data set but also is common in other domain text translations like a software manual. We aim at investigating when to and when not to translate from scratch in such scenarios.

3.1 Retrieval Approaches on Translation Memory

The retrieval approach on the translation memory is useful, since it is well known that NMT models are data-hungry and it is difficult to control the translation outputs generated by NMT models. At inference time, we calculate a sentence-level similarity score between a query, i.e. a given source sentence, and all the source sentences stored in the translation memory. If there exists a source sentence in the translation memory whose similarity score is above the threshold, we employ its target sentence as a final output. In our systems, we provide two types of retrieval approaches: 1) Edit-distance-based retrieval and 2) Inverse document frequency(IDF)-based retrieval.

Edit-distance-based retrieval The edit-distance-based retrieval is a widely-used method in work that investigates using translation memories to enhance NMT models (Gu et al., 2018; Cao and Xiong, 2018). We calculate the similarity score between two source sentences (S_1 and S_2) using the character-based Levenshtein distance as follows:

$$\mathrm{Sim}_{edit}(S_1, S_2) = 1 - \frac{\Delta_{dist}(S_1, S_2)}{\max\left(|S_1|, |S_2|\right)}, \quad (1)$$

where Δ_{dist} indicates the Levenshtein distance of sentences S_1 and S_2. $|S|$ denotes the length of a sentence S.

IDF-based retrieval An IDF-based retrieval approach was investigated by Bapna and Firat (2019). Following the previous work, we calculate a sentence-level similarity score by using an IDF score f_t of a token t as follows:

$$\mathrm{Sim}_{idf}(S_1, S_2) = 2 \times \sum_{t \in (S_1 \bigcup S_2)} f_t - \sum_{t \in (S_1 \bigcap S_2)} f_t, \quad (2)$$

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

$$f_t = \log \frac{|C_{TM}|}{n_t}, \quad (3)$$

where $|C_{TM}|$ is the number of sentence pairs in the translation memory. n_t denotes the number of occurrences of a token t in the corpus. In our preliminary experiments, we found that using sub-words for a token unit t is better than characters. We also tried IDF-based n-gram retrieval proposed in (Bapna and Firat, 2019); however, the two above-mentioned retrieval methods always worked better.

3.2 Neural Machine Translation

We employ a Transformer (base) model (Vaswani et al., 2017) as a default NMT system in our proposed approach. Transformer is modeled as an encoder-decoder network architecture, where an input sentence $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is encoded into a fixed vector space and decoded from the fixed vector to the output sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$. Following Vaswani et al. (2017), the inputs are mapped into the 512-dimensional embedding space with positional embedding. Both the encoder and decoder networks map the vectors through 6-layer 2048-dimensional feed-forward networks with 8-head self-attention and layer-normalization (Ba et al., 2016), and the decoder has an 8-head attention layer before the feed-forward network layer between the target hidden state and the source hidden states. We shared the parameters across the target embeddings and a softmax layer in the decoder (Inan et al., 2017; Hashimoto and Tsuruoka, 2017). To avoid overfitting, we use dropout with the rate of 0.1 and introduce the label-smoothed cross entropy loss with the coefficient of 0.1 (Pereyra et al., 2017). We use Adam (Kingma and Ba, 2015) to optimize all model parameters. We apply warm-up learning rate scheduling, increasing the learning rate linearly during predefined warm-up updates and applying learning rate decay based on the inverse square root of the update number (Vaswani et al., 2017).

4 Experimental Design

Data Preparation All of the training corpora provided for ITEM and TEXT data are concatenated into a single training corpus. We also use the 1M Japanese-English Wikipedia parallel corpus provided by Asai et al. (2018) as an additional training resource. The corpus is automatically created by crawling multilingual Wikipedia pages and applying a sentence aligner. Because the parallel data in that corpus are pre-tokenized, we applied a detokenization script on both sides. In preliminary experiments, we confirmed that using the additional Wikipedia training data improved translation accuracy on the task.

All of the data sets are tokenized using `SentencePiece` (Kudo and Richardson, 2018), and we set the vocabulary size to 32k. To determine the optimal sentence-similarity threshold in the retrieval approaches, we evaluated the systems based on `sacreBLEU` score (Post, 2018) with thresholds varying within $\{80, 100\}$ and $\{10, 25\}$ for THRESHOLD$_{edit}$ and THRESHOLD$_{IDF}$, respectively. The best thresholds for each NMT system are determined based on the development results. When tuning our submission systems, we perform the threshold optimization on the devtest data, and the development data is added into the translation memory.

System Description A (Marian) We use a codebase of *Marian* (Junczys-Dowmunt et al., 2018) to train the Transformer model described in Subsection 3.2. In System A, we set the mini-batch size to 1,000. The initial learning rate and warm-up steps are set to 0.0002 and 8,000. The maximum length of the training examples is set to 100, and 0.4% training data are discarded during the training. We trained the system for 200k updates with 8 GPUs. Regarding data preprocessing, we create a joint vocabulary with the size of 32k. We refer System A as "Marian" after this.

System Description B (Fairseq) We use a vocabulary set separately created in the source and target languages, and each vocabulary size is set to 32k. We fill a mini-batch with up to 6,000 tokens, and we use the initial learning rate of `1e-07` and warm-up updates of 2,000. We trained the model for 80k updates with 4 GPUs. We use a codebase of *Fairseq* (Ott et al., 2019). We refer System B as "Fairseq" in the following sections. At inference time, we use the beam-search decoding with the size of $\{4, 8, 12\}$ and select the best beam size based on the development results for both systems.

Large-scale Black-box MT systems To verify the effectiveness of using translation memory on the task, we experiment by using three types of production-level black-box MT systems, i.e.

	Dev.	Devtest
Marian	48.6	50.6
Fairseq	40.9	42.2
Online A	24.8	24.8
Online B	24.5	24.4
Online C	24.5	24.5

Table 2: General translation accuracy of each system on the concatenated data (ITEM+TEXT).

Google Translate[2], Microsoft Bing Translator[3], and Mirai Translate[4]. More concretely, we replace our MT outputs with those of the production MT systems and evaluate the translation performance. These online MT systems are anonymized into Online A, Online B and Online C in a random order after this section.

5 Results and Discussion

5.1 Experimental Results

First of all, we evaluate the overall translation accuracy of each NMT system and production system on the concatenated data (ITEM+TEXT). Table 2 reports the case-sensitive sacreBLEU scores of the NMT systems and the production systems without translation memory. Marian shows the best BLEU score on both development and devtest datasets. The reason for this can be that Marian as an over-trained model translates better on the duplicates. We also see huge gaps between our white-box NMT systems and online black-box MT systems at least by 16.1 and 17.4 BLEU scores on development and devtest data sets, respectively. The three online systems show equivalent accuracy with each other due to lacking the training examples of the task.

Table 3 shows the experimental results of our proposed approach using the translation memory, reporting the sacreBLEU scores on the ITEM and TEXT evaluation data sets. The best retrieval approach is different for those two data sets. We found that in general, edit-distance-based retrieval produces better results for ITEM data, while the IDF-based retrieval works better for TEXT data. The only exception to this was Online C on TEXT. Higher sentence-similarity thresholds were selected for Marian and Fairseq with the ITEM data, indicating that the outputs generated by these systems show better quality than those by the online systems. Introducing the translation memory to the systems outputs, however, we can largely fill the gaps between our systems and the online systems by around 1-2 BLEU scores on both evaluation data sets.

		threshold	Dev.	Devtest
ITEM	Marian	89	54.1	58.1
	Fairseq	89	53.3	57.0
	Online A	83	51.2	55.6
	Online B	80	51.8	55.8
	Online C	83	51.6	55.6
TEXT	Marian	18	57.7	57.9
	Fairseq	14	57.1	57.6
	Online A	15	55.9	56.7
	Online B	10	55.6	56.4
	Online C	80	55.7	56.8

Table 3: BLEU results of our proposed approach on the evaluation data sets (ITEM and TEXT).

		threshold	Devtest.	Test
ITEM	Marian	95	57.5	54.27
	Fairseq	89	56.6	50.90
TEXT	Marian	21	58.0	61.38
	Fairseq	18	57.7	51.08

Table 4: BLEU results on the evaluation data sets (ITEM and TEXT). We employed edit-distance-based and IDF-based retrieval approaches for evaluation on the ITEM and TEXT data sets, respectively. The BLEU scores on the test set are cited from the official leader board (http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/).

Table 4 shows the results of our submission systems on the devtest and test data sets, where Fairseq provides the result of an ensemble with 4 replicas. We include the development translation pairs in the translation memory, and selected the threshold to use on the test data based on scores on the devtest data set.

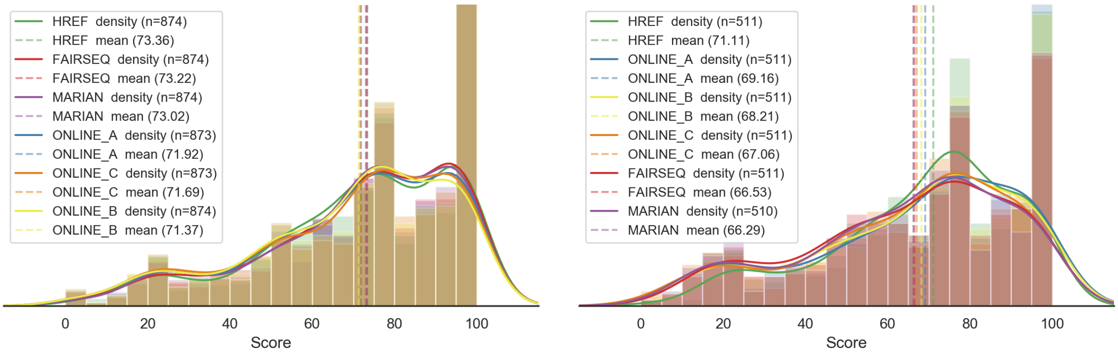
5.2 Human Evaluation

We used source-based direct assessment for human evaluation, as described in Cettolo et al.

[2] https://translate.google.com/, as of July, 2019.

[3] https://www.bing.com/translator, as of July, 2019.

[4] https://miraitranslate.com/en/ as of July, 2019.



(a) ITEM Score Distribution

(b) TEXT Score Distribution

Figure 2: Human Evaluation Score Distributions

(2017). For the annotation process, we used an updated version of Appraise (Federmann, 2012), the human evaluation tool used for the Conference on Machine Translation (WMT)[5], and we followed the evaluation campaign setup as specified in Hassan et al. (2018). In source-based direct assessment, annotators are shown source text and a candidate translation and are asked the question "How accurately does the above candidate text convey the semantics of the source text?", answering this using a slider ranging from 0 (Not at all) to 100 (Perfectly).

In this campaign we examined five systems: Marian, Fairseq, Online A, Online B and Online C. We also added human reference (HREF) to the campaign for comparison. Table 5 shows the evaluation campaign parameters. We hired 25 bilingual crowd-sourced annotators and assigned two tasks to each. We collected a single score for each of the randomly selected translations on each system. There were no overlapping annotation items among annotators. For the ITEM testset, we obtained at least 873 assessments for each system[6]. Likewise we obtained 511 assessments for the TEXT testset. We collected a total of 8,308 annotation data points.

Table 6 shows the mean scores for ITEM, TEXT and ALL (ITEM+TEXT) for each system. A boldfaced number indicates that the mean score is indistinguishable from HREF in the same category (ITEM, TEXT or ALL) using the Mann-Whitney U Test at p-level $p < 0.05$. Figures 2a and 2b show the evaluation score distributions for ITEM and TEXT, respectively.

Testset:	Devtest
Annotators:	25
Tasks:	50
Redundancy:	1
Task per Annotator:	2
Data points:	8308

Table 5: Human Evaluation Campaign Parameters

	ITEM	TEXT	ALL
HREF	73.4	71.0	72.5
Fairseq	**73.2**	66.5	70.8
Marian	**73.0**	66.3	70.5
Online A	**71.9**	69.2	**70.9**
Online B	**71.4**	68.2	70.2
Online C	**71.7**	67.1	70.0

Table 6: Human Evaluation results. The boldfaced numbers indicate that they are indistinguishable from HREF at p-level $p < 0.05$.

5.3 Discussion

The human evaluation results indicate that the translation quality of each system is comparable with human reference for the ITEM testset while the differences were statistically significant for the TEXT testset. One possible reason for this is that the retrieval approach using the translation memory works better for shorter sentence translation in the ITEM dataset but not for longer sentence translation in the TEXT dataset. The average English sentence length of the ITEM devtest is 7.7, whereas that of the TEXT devtest is 25.6. Regarding the number of unique words, the ITEM data contains 22,453 vocabulary items,

[5] http://www.statmt.org/wmt19/

[6] We obtained 874 assessments for some systems.

	#	ITEM
Source	—	依頼者提示資料に基づき査定
HREF	28	Based on materials provided by IIA
Fairseq	99	Assessed based on documents presented by the requester.
Marian	99	Assessed based on documents presented by the requester.
Online A	99	Assessed based on documents presented by the requester.
Online B	99	Assessed based on documents presented by the requester.
Online C	99	Assessed based on documents presented by the requester.

Table 7: Translation examples of each system on the devtest data set (ITEM) that obtain a higher evaluation score than the human reference. “#” denotes the human evaluation score. All the translations are retrieved from the translation memory.

	#	TEXT
Source	—	なお、当社は平成31年3月期の配当予想を年間配当金62円00銭といたしました。
HREF	97	In addition, the Company made an annual dividend of 62.00 yen for the fiscal year ending March 31, 2019.
Fairseq	96	The dividend forecast for the fiscal year ending March 31, 2019 is 62.00 yen per share.
Marian	95	The Company’s dividend forecast for the fiscal year ending March 31, 2019 is projected to be ¥62.00 per share.

Table 8: Translation examples of each system on the devtest data set (TEXT) that are highly evaluated by human annotators. The column of “#” reports the human evaluation score of each output. Both translation outputs are generated by the NMT models.

while the TEXT datasets does 28,507. These different trends between the ITEM and TEXT data suggest that the systems are required to translate relatively fixed phrases or sentences more in the ITEM data set, which is a suitable scenario for translation memories. On the other hand, it also suggests that MT is more desirable in the cases where long sentences with a variety of expressions need to be translated. However, this is not always the case for the TEXT translations with our systems because longer sentences which may contain major semantic errors can be chosen due to their high similarity scores. Tables 7 and 8 show each system’s translation outputs and its human evaluation score on the devtest data set (ITEM and TEXT).

Our approach of combining a translation memory with MT systems is evaluated lower than the human reference by human annotators on the TEXT data, whereas the online systems are more highly evaluated among all the systems. It is because those production systems are better at translating longer sentences due to much larger training corpora. For instance, Google Translate is trained on three or four orders of magnitudes larger training data (Johnson et al., 2017), and such a system should cover a variety of expressions and domains. Thus, it is still important to improve the translation quality of NMT models on longer sentences, which has been actively studied in the context of document-level translation (Jean et al., 2017; Tiedemann and Scherrer, 2017; Junczys-Dowmunt, 2019).

6 Conclusion

This paper describes our submission systems to WAT’19 Timely Disclosure task. The system is a combination approach of translation memory and NMT model. First, we observed that 26-28% of data are duplicated between training data and test sets. The system enables us to directly retrieve a translation candidate from the translation memory. Any MT model can be applied to our approach, and we confirmed its effectiveness even when using black-box MT production systems. Results from the human evaluation campaign demonstrate that translation on a fixed form or short expressions can be covered well with translation memory, while NMT is much more robust especially when flexible translation on longer sentences is required.

References

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual extractive reading comprehension by runtime machine translation. In *arXiv preprint arXiv:1809.03275*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. In *arXiv preprint arXiv:1607.06450*.

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Qian Cao and Deyi Xiong. 2018. Encoding gated translation memory into neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3042–3047.

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

Christian Federmann. 2012. Appraise: an open-source toolkit for manual evaluation of MT output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search engine guided neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pages 5133–5140.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2017. Neural machine translation with source-side latent graph parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 125–135, Copenhagen, Denmark.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic Chinese to English news translation. *arXiv preprint arXiv:1803.05567*.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. *In 7th International Conference on Learning Representations*.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arxiv preprint arXiv:1704.05135*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

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, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Philipp Koehn and Jean Senellart. 2010. Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.

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.

Toshiaki Nakazawa, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Nobushige Doi, Yusuke Oda, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2019. Overview of the 6th workshop on Asian translation. In *Proceedings of the 6th Workshop on Asian Translation*, Hong Kong. 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*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *In Proceedings of International Conference on Learning Representation Workshop*.

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.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Yang Zhao, Yining Wang, Jiajun Zhang, and Chengqing Zong. 2018. Phrase table as recommendation memory for neural machine translation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4609–4615. International Joint Conferences on Artificial Intelligence Organization.