NTT Neural Machine Translation Systems at WAT 2019

Makoto Morishita[*], **Jun Suzuki**[*] and **Masaaki Nagata**
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
{makoto.morishita.gr, masaaki.nagata.et}@hco.ntt.co.jp
jun.suzuki@ecei.tohoku.ac.jp

Abstract

In this paper, we describe our systems that were submitted to the translation shared tasks at WAT 2019. This year, we participated in two distinct types of subtasks, a scientific paper subtask and a timely disclosure subtask, where we only considered English-to-Japanese and Japanese-to-English translation directions. We submitted two systems (En-Ja and Ja-En) for the scientific paper subtask and two systems (Ja-En, texts, items) for the timely disclosure subtask. Three of our four systems obtained the best human evaluation performances. We also confirmed that our new additional web-crawled parallel corpus improves the performance in unconstrained settings.

1 Introduction

We participated in a scientific paper subtask and a timely disclosure subtask at this year's shared translation tasks at WAT 2019 (Nakazawa et al., 2019). Since we only considered English-to-Japanese (**En-Ja**) and Japanese-to-English (**Ja-En**) translation directions, we submitted En-Ja and Ja-En systems for the scientific paper subtask and two Ja-En systems (texts, items) for the timely disclosure subtask. The base NMT model architecture that we employed is a widely used Transformer model, but we tried to explore a better set of hyper-parameters, leading to significant improvement. Three of our submissions were honored as the best human evaluation performances. As our new trial, we evaluated the usefulness of incorporating external data automatically collected from a wide variety of web pages to further improve the translation quality.

We independently developed two distinct systems for each subtask. Therefore, this paper separately explains the details; we first explain the systems developed for the scientific paper subtask in Section 2. Then we describe the system developed for the timely disclosure subtask in Section 3.

Set	# Sentences
Train	3,008,500
(bitext)	(1,500,000)
(synthetic)	(1,508,500)
Dev	1,790
Devtest	1,784
Test	1,812

Table 1: Numbers of sentences in ASPEC corpus

2 Systems for Scientific Paper Subtask

2.1 Task Overview

For the scientific paper task, we participated in two translation directions: Japanese-to-English (Ja–En) and English-to-Japanese (En–Ja). We submitted two systems per direction in two different training settings: constrained and unconstrained settings.

2.2 Data and Data Preparation

2.2.1 Provided data: constrained setting

As training/dev/test data, the task organizer provided the Asian Scientific Paper Excerpt Corpus (ASPEC) (Nakazawa et al., 2016) whose statistics are shown in Table 1.

ASPEC was created by automatically aligning parallel documents and sentences, and the training sentences are ordered by sentence alignment scores. Thus, the previous participants generally removed the latter sentences (Neubig, 2014) or used them as synthetic data (Morishita et al., 2017). This year, we used the former 1.5M training sentences as bitext data and the latter 1.5M as monolingual data and created synthetic data (Sennrich et al., 2016).

[*] Equal contribution.

Proceedings of the 6th Workshop on Asian Translation, pages 99–105
Hong Kong, China, November 4, 2019. ©2019 Association for Computational Linguistics

2.2.2 JParaCrawl: unconstrained setting

The ParaCrawl[1] project is building parallel corpora by largely crawling the web. Their objective is to build parallel corpora for the 24 official languages of the European Union. They already released earlier versions of the corpora and they were used on the WMT 2018 news shared translation tasks (Bojar et al., 2018). The WMT shared task participants reported that this corpora boosted translation accuracy when used with careful corpus cleaning (Junczys-Dowmunt, 2018; Morishita et al., 2018).

Inspired by these previous works, we constructed a web-based Japanese-English parallel corpus. We followed almost the same procedure as ParaCrawl to make this corpus. First, we listed 100,000 candidate domains that might contain parallel Japanese and English sentences by analyzing the whole Common Crawl text data[2] on how each domain contains Japanese or English data with `extractor`[3].

We crawled the listed candidate domains and aligned parallel sentences using `bitextor`[4]. Then we filtered out noisy sentences with `bicleaner`[5] (Sánchez-Cartagena et al., 2018). After corpus cleaning, we retained 7.5M sentences.

We named this corpus "JParaCrawl" and we plan to release it publicly with a detailed corpus description paper.

2.2.3 Data Preprocessing

This year, we decided not to employ any external morphological analyzer like KyTea (Neubig et al., 2011). Instead we utilized `sentencepiece`[6] (Kudo and Richardson, 2018), which tokenizes a sentence into a sequence of subwords without requiring any other tokenizers. Note here that we did not apply any filtering method, such as sentence length filtering.

2.3 System Details

We selected the Transformer model (Vaswani et al., 2017) as our base NMT model. We also incorporated two techniques to further improve the performance: (1) model ensembling and (2) Right-to-Left (R2L) re-ranking.

2.3.1 Ensembling

We independently trained four models with different random seeds and simultaneously utilized them for model ensembling to boost the translation performance.

2.3.2 Right-to-left (R2L) re-ranking

The NMT model has auto-regressive architecture in its decoder that uses previously generated tokens for predicting the next token. In other words, we normally decode a sentence from the beginning-of-the-sentence (BOS), which is its left side, to the end-of-the-sentence (EOS), which is on the right. Here we call this normal decoding process as Left-to-Right (L2R) decoding. However, Liu et al. (2016) pointed out that L2R decoding lacks reliability near the EOS tokens because if the previous tokens contain errors, the next prediction might have error as well. To alleviate this problem, Liu et al. (2016) proposed a method that generates the n-best hypotheses with the L2R model and re-ranks them with the R2L model, which decodes the sentences from the EOS tokens to the BOS tokens. By R2L re-ranking, we can exploit both the advantages of the L2R and R2L models and improve their performance.

2.3.3 Model incorporation with JParaCrawl

The JParaCrawl domain basically differs from the scientific paper task. To effectively incorporate with the JParaCrawl, we first pre-trained the model with the mixed data of ASPEC and JParaCrawl.[7] Then we fine-tuned the pre-trained model using only ASPEC.

2.4 Hyper-parameter

As a base NMT model, we selected the Transformer model with the "big" hyper-parameter setting. During training, we used mixed-precision training (Micikevicius et al., 2018) that can boost the training speed and reduce the memory consumption. We saved the model each epoch and used the average of the last ten models for decoding. We set the beam size to six and normalized the scores by their length. All implementations are based on `fairseq` toolkit (Ott et al., 2019). Table 2 shows the selected set of the

[1] http://paracrawl.eu/

[2] https://commoncrawl.org/

[3] https://github.com/paracrawl/extractor

[4] https://github.com/bitextor/bitextor

[5] https://github.com/bitextor/bicleaner

[6] https://github.com/google/sentencepiece

[7] We mixed ASPEC and JParaCrawl by upsampling ASPEC twice.

Hyper-parameter	Selected Value
Subword (vocabulary) size	src:4000, trg:4000
Gradient clipping	1.0
Dropout rate	0.3
Mini-batch size	4K tokens
Update frequency	128 batches
Beam search (n-best)	6 best

Table 2: Hyper-parameters for the scientific paper task: Our basic hyper-parameters are identical to Transformer "big" setting.

Subword size	En-Ja	Ja-En
w/o synthetic data	44.2	29.9
Back-translation	**45.6**	29.5
Forward-translation	–	**30.1**

Table 3: Comparison of translation performance on changing subword size. Scores here were calculated by `sacreBLEU`.

hyper-parameters we used for the final submission. In our preliminary experiments, we evaluated extensive combinations of hyper-parameters and we found that this setting was optimal in our hyper-parameter search. Hereafter, the reported performance in the rest of this paper was obtained using this setting unless otherwise specified.

2.4.1 Back- and forward-translation for building synthetic data

We first investigated the effectiveness of incorporating synthetic data generated by the back-translation technique. Table 3 shows the results. We significantly improved the performance of the En-Ja translation setting by adding the synthetic data. However surprisingly, the performance was significantly degraded (29.9 → 29.5) in the Ja-En translation setting. We observed that the quality of the English sentences in the latter half of the provided data looks somewhat awful (not very well). Therefore, we then tried to make synthetic data by using forward-translation instead of the standard back-translation. This means that we used the synthetic data for the En-Ja translation setting as the synthetic data of the Ja-En translation setting. This slightly improved the performance of the Ja-En translation setting.

2.4.2 Subword size/Vocabulary size

Table 4 shows the BLEU scores when we changed the number of subwords obtained from `sentencepiece`. Note that we evaluated the performance using `sacreBLEU` (Post, 2018) for all the results shown in this section.

Subword size	En-Ja	Ja-En
4000	**45.6**	**30.1**
8000	45.3	29.9
16000	45.2	29.6
32000	45.0	29.7

Table 4: Comparison of translation performance on changing the methods of building synthetic data. Scores here were calculated by `sacreBLEU`.

Mini-batch size	En-Ja	Ja-En
16 × 4,000 tokens	45.1	29.7
32 × 4,000 tokens	45.3	29.9
64 × 4,000 tokens	45.4	29.8
128 × 4,000 tokens	**45.6**	**30.1**
256 × 4,000 tokens	45.4	29.9

Table 5: Comparison of translation performance on changing mini-batch size (update frequency) for each update in NMT training. Scores here were calculated by `sacreBLEU`.

We clearly observe a tendency that the fewer subwords got better performance. This observation is actually a bit surprising since many recent previous studies in the NMT community often employed a larger amount of subwords like 16,000 or 32,000.

2.4.3 Mini-batch size/Update frequency

According to a previously introduced finding, Transformer models tend to provide better results with a larger mini-batch size (Ott et al., 2018). Based on this observation, we explored the effectiveness of the mini-batch size in our setting.

Table 5 shows the results. We found that an overly large mini-batch, i.e., 512, degraded the performance. In our experiments, an update frequency of 128, which means $128 \times 4,000 = 512,000$ tokens per mini-batch, was an appropriate value.

2.4.4 Ensemble and R2L re-ranking

Ensembling and re-ranking are currently the standard techniques for further improving the translation quality in the NMT models. Following this public knowledge, we also applied standard ensembling and right-to-left (R2L) re-ranking techniques to our models.

Table 6 shows the effectiveness of these techniques. Ensembling and R2L re-ranking offered

Model type	En-Ja	Ja-En
Single model (equivalent to 1)	45.6	30.1
Ensemble (4)	46.2	30.8
Ensemble (4) + R2L (4)	46.8	**31.2**
Ensemble (6) + R2L (4)	**46.9**	**31.2**

Table 6: Results of incorporating ensembling and R2L re-ranking techniques. The numbers in brackets shows the number of models for ensembling, e.g., (4) masn four model ensembling. Scores here were calculated by `sacreBLEU`.

Data	En-Ja	Ja-En
Ensemble (4) + R2L (4)		
(ASPEC only)	46.8	31.2
(ASPEC+JParaCrawl)	**47.4**	**31.6**

Table 7: Translation performance comparison when we incorporate additional training data JParaCrawl. Scores here were calculated by `sacreBLEU`.

significant improvements.

2.4.5 Unconstrained setting

Table 7 shows the "Ensemble (4) + R2L (4)" results that were trained by ASPEC or ASPEC+JParaCrawl.

Incorporating JParaCrawl consistently and significantly improved performance . This fact indicates that using more data improves better performance; even the additional data (JParaCrawl) domain slightly differs from the target domain.

2.5 Official Result

We first planned to submit the *unconstrained setting* results (the second row in Table 7) as our primary results. Unfortunately, we failed to finish training for all the models (four L2R and four R2L models) by the submission deadline. Therefore, we submitted the *constrained setting* results (the first row in Table 7) as our primary results.

Table 8 shows the official results of our submissions computed in the evaluation server. Our system achieved the best BLEU score for the En-Ja subtask, but slightly lower than the best system for the Ja-En subtask. For pairwise crowd-sourcing evaluations, our system successfully obtained the best assessments for both the En-Ja and Ja-En subtasks. Our system also achieved the best performance in terms of adequacy for the En-Ja subtask. Although our Ja-En system ranked second, the gap between both systems is quite small (0.02).

2.6 Post-evaluation

As described in the previous section, since we could not finish training the unconstrained setting by the submission deadline, we evaluated the results of the unconstrained setting in the evaluation server as a post-evaluation. Table 9 shows the results. We further improved the official best scores for both the En-Ja and Ja-En subtasks: +0.74 for En-Ja and +0.67 for Ja-En.

3 Systems for Timely Disclosure Subtask

3.1 Task Overview

The new timely disclosure task focuses on translating Japanese company's announcements for investors into English. It is challenging because the documents contain many figures and proper nouns that are critical but difficult to translate.

The provided corpus sizes are shown in Table 10. This task has two sub-tasks: *texts* and *items*. The *texts* task contains the sentences whose Japanese side ends with "。" (Japanese period), and the *items* includes subjects, table titles, and bullet points.

Note that the data provider releases a detailed corpus description[8] that includes the corpus characteristics, the text normalization rules, and how they separate the data into *texts* and *items*. This description was quite useful when we tackled the task.

3.2 System Details

Our submission includes three features: (1) task-specific fine-tuning, (2) right-to-left re-ranking, and (3) model ensembles.

As mentioned in Section 3.1, this task has been separated into two categories: *texts* and *items*. Although the provided training data were not split, we easily separated them into sub-categories by just checking whether the Japanese sentence ends with "。" or not. To achieve the best performance, we first pre-trained the model with all of the provided training data and fine-tuned it with the specific parts of the training data. We also use the ensembling and R2L re-ranking techniques, as described in Sections 2.3.1 and 2.3.2. During the R2L re-ranking, we ensembled the R2L models in addition to the L2R models for better performance.

[8] `http://lotus.kuee.kyoto-u.ac.jp/WAT/Timely_Disclosure_Documents_Corpus/specifications_en.html`

Lang.	Auto Eval		Human Eval			
pair	BLEU	(Rank)	Pairwise	(Rank)	Adequacy	(Rank)
En-Ja	45.83	(1)	47.75	(1)	4.50	(1)
Ja-En	30.56	(5)	14.00	(1)	4.49	(2)

Table 8: Official results of our submitted systems for ASPEC subtask: For En-Ja direction, we show BLEU scores with JUMAN tokenizer.

Training data	En-Ja	Ja-En
ASPEC	45.83	30.56
ASPEC+JParaCrawl	**46.57**	**31.23**

Table 9: Performance comparison when we incorporate additional training data JParaCrawl: Scores here were obtained from evaluation server.

Set	Category	# Sentences
Train	texts	448,472
	items	955,523
Dev	texts	1,153
	items	2,845
Devtest	texts	1,114
	items	2,900
Test	texts	1,148
	items	2,129

Table 10: Number of sentences in timely disclosure document corpus: We split training set into two categories. See Section 3.2 for details.

3.3 Experimental Settings

For preprocessing, we only relied on `sentencepiece` (Kudo and Richardson, 2018), which tokenizes a sentence into subwords without requiring any other tokenizers. We set the vocabulary size to 32k[9]. The provided training data were split by their released years, but we concatenated them without distinguishing them.

As an NMT model, we used the Transformer (Vaswani et al., 2017) with big hyper-parameter settings and dropout (Srivastava et al., 2014) with a probability of 0.3. We trained the model with eight RTX 2080 Ti GPUs and set a batch size of 2,500 tokens. We accumulated 128 mini-batches per update (Ott et al., 2018), resulting in a per-update batch size around $128 \times 2,500 = 320,000$ tokens. Based on the validation perplexity, we stopped the training when the update count reached 5,000 and fine-tuned the model for 800 updates. During training, we used mixed-precision training (Micikevicius et al., 2018), like the scientific paper subtasks. We saved the model every 100 updates and used the average of the last eight models for decoding. We set the beam size to six and normalized the scores by length. All implementations are based on `fairseq` toolkit (Ott et al., 2019).

	Texts	Items
Baseline model	55.26	54.58
+ Fine-tune	58.91	56.14
+ Ensemble (4 models)	60.48	56.93
+ R2L re-ranking (4 models)	**61.19**	**57.34**

Table 11: Case-sensitive BLEU scores of provided blind test sets: All scores were calculated by official evaluation server.

3.4 Experimental Results and Analysis

Table 11 shows the case-sensitive BLEU scores (Papineni et al., 2002) of the provided blind test sets. All the reported BLEU scores were calculated on the organizers' submission website.

3.4.1 Baseline model

We set the baseline system as a single NMT model trained with all the training data. Note that we used the same model for both categories in the baseline setting. Even the baseline system achieved around 55 points in both categories. This means that the model already outputs quite similar hypotheses as references.

3.4.2 Fine-tuning with a specific category

We found that fine-tuning with specific category data significantly increased the BLEU scores: $+3.65$ points for texts and $+1.56$ points for the items categories. Table 13 shows the example translations of the baseline and fine-tuned systems[10]. The fine-tuned system perfectly gener-

[9] In contrast to the scientific paper subtasks, we did not see improvement with a smaller vocabulary in the preliminary experiments.

[10] For finding good examples, we used `compare-mt` (Neubig et al., 2019), which is a toolkit that compares two MT outputs.

	Auto Eval		Human Eval			
Task	BLEU	(Rank)	Pairwise	(Rank)	Adequacy	(Rank)
Texts	61.19	(1)	55.50	(1)	4.46	(1)
Items	57.34	(1)	34.00	(2)	4.47	(1)

Table 12: Official results of our submitted systems for timely disclosure subtask: Shown rank is only ordered among constrained submissions.

Input	実績値、類似建物の修繕費水準、エンジニアリング・レポートの修繕更新費等を考慮し査定
Reference	Based on historical data, comparable assets and estimates in the engineering report
Baseline	Assessed by taking into account the actual results, the level of repair expenses of similar buildings, the level of repair expenses in the engineering report, etc.
Fine-tuned	Based on historical data, comparable assets and estimates in the engineering report

Table 13: Example translations of baseline and fine-tuned system: Example was picked from devtest set.

Japanese	実績値、類似建物の修繕費水準、ERの修繕更新費等を考慮し査定
English	Based on historical data, comparable assets and estimates in the engineering report

Table 14: Example of sentence pair contained in the training set.

ated the same sentence as the reference. Although the baseline's hypothesis is also understandable, it does not match the *items* context. We further investigated why the fine-tuned system works so well, and we suspected that the sentences in the dev/test set mostly overlap with the training set; i.e. it might be possible to find almost the same sentence from the training set. Table 14 shows the sentence pair in the training set that was the most similar to the previous example. We found a sentence pair on the English side that is identical as the reference in Table 13, and the Japanese side is also quite similar. By fine-tuning, the model is somewhat over-fitted to the specific categories and memorized more training sentences. Thus, in this case, fine-tuning provides a large gain.

3.4.3 Ensemble and R2L re-ranking

Model ensembling and R2L re-ranking also improved the scores. Additional gains from both are $+2.28$ points for texts and $+1.20$ points for items compared with the fine-tuned models.

3.4.4 Submissions and human evaluations

Table 12 shows our submissions and their human evaluation scores. We achieved the best scores in terms of BLEU for both subtasks among the constrained submissions. By pairwise evaluation, our submission to the text subtask ranked first and the items subtask ranked second. However, in the adequacy evaluations, our system achieved top performance in both the texts and items subtasks.

4 Conclusion

We described the systems we submitted to the WAT 2019 shared translation tasks. We submitted the systems for scientific translation subtasks and timely disclosure subtasks and three of four systems won the best human evaluation performance. We also confirmed that an additional web-crawled based parallel corpus increased the performance on the scientific paper subtasks.

Acknowledgement

We thank the ParaCrawl project for their contribution and releasing the software, which we used for creating JParaCrawl.

References

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 272–303.

Marcin Junczys-Dowmunt. 2018. Microsoft's submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 425–430.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 66–71.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 411–416.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In Proceedings of the 6th International Conference on Learning Representations (ICLR).

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2017. NTT neural machine translation systems at WAT 2017. In Proceedings of the 4th Workshop on Asian Translation (WAT), pages 89–94.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. NTT's neural machine translation systems for WMT 2018. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 461–466.

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

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC), pages 2204–2208.

Graham Neubig. 2014. Forest-to-string SMT for asian language translation: NAIST at WAT2014. In Proceedings of the 1st Workshop on Asian Translation (WAT), pages 20–25.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. comparemt: A tool for holistic comparison of language generation systems. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 35–41.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), pages 529–533.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 48–53.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 1–9.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 311–318.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 186–191.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez-Sánchez. 2018. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 955–962.

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 (ACL), pages 86–96.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958.

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 Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), pages 6000–6010.