The University of Tokyo's Submissions to the WAT 2020 Shared Task

Matīss Rikters and Ryokan Ri and Toshiaki Nakazawa
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
{matiss, li0123, nakazawa}@logos.t.u-tokyo.ac.jp

Abstract

The paper describes the development process of the The University of Tokyo's NMT systems that were submitted to the WAT 2020 Document-level Business Scene Dialogue Translation sub-task. We describe the data processing workflow, NMT system training architectures, and automatic evaluation results. For the WAT 2020 shared task, we submitted 26 systems (both with and without using other resources) for English-Japanese and Japanese-English translation directions. The submitted systems were trained using Transformer models and one was a SMT baseline.

1 Introduction

We describe the machine translation (MT) systems submitted to the WAT 2020 Document-level Business Scene Dialogue Translation sub-task developed by the team of The University of Tokyo. We chose the identifier of our team to be ut-mrt, which specifies our affiliation (The University of Tokyo) and first names (Matīss, Ryokan, Toshiaki). We participated in both EN→JA and JA→EN translation directions. We experimented with mixing and matching several data sets, data processing approaches and training methods.

Our main findings are: 1) using source side context mainly improves EN→JA MT, but not always, and mainly degrades or leaves little impact on JA→EN MT; 2) there are no better data than more data - we see the biggest improvements from using larger training data sets; and 3) optimiser delay (Bogoychev et al., 2018) can help a lot - by setting the optimiser delay value to 8 instead of the default 1 increased BLEU scores by more than 1.5 in both translation directions.

2 Data

We used multiple dataset combinations to train our models for the shared task. We also filtered some of the larger automatically collected data sets which are usually more noisy and contain duplicates.

Aside from using only the provided BSD training dataset (BSD 20 (Rikters et al., 2019)), we had access to an extended version of the BSD four times the size (BSD 80), as well as two other similar corpora - AMI Meeting corpus (AMI) and a parallel version of OntoNotes 5.0 (ON) (Rikters et al., 2020). We also experimented with using the jParaCrawl (Morishita et al., 2019) dataset, data from WMT 2020[1] (whcih includes JParaCrawl, Ted Talks (Cettolo et al., 2012), The Kyoto Free Translation Task Corpus (Neubig, 2011), Japanese-English Subtitle Corpus (Pryzant et al., 2018), WikiMatrix (Schwenk et al., 2019) and Wiki Titles v2), and a proprietary document-aligned news dataset gathered from several sources. The full training data statistics are shown in Table 1. The AMI and jParaCrawl corpora contain many duplicates while the rest seem to be of higher quality.

	Total	Unique	Filtered
BSD 20	20,000	18,818	17,672
BSD 80	80,629	74,377	69,742
AMI	110,483	75,660	57,046
ON	28,429	24,335	18,348
WMT	17,880,587	16,501,296	13,035,839
jParaCrawl	10,105,351	8,790,618	7,087,631
News	1,104,549	1,101,751	956,654

Table 1: Total, unique data amounts and after filtering for the noisiest corpora.

2.1 Filtering

We used data filtering methods described by Rikters (2018) to remove the noisiest parts of the corpora for experiments involving jParaCrawl. The filtering process consists of the following filters: 1) unique parallel sentence filter, which removes duplicate parallel sentences; 2) equal source-target

[1] http://www.statmt.org/wmt20/translation-task.html

Proceedings of the 7th Workshop on Asian Translation, pages 147–153
December 4, 2020. ©2020 Association for Computational Linguistics

filter, which removes parallel sentences that are identical in both languages; 3) multiple sources - one target and multiple targets - one source filters; 4) non-alphabetical filters - remove sentences having a majority of characters outside the scope of the specified language; 5) repeating token filter, which removes sentences that have several repeating tokens or phrases in a row; and 6) correct language filter, which uses language identification (Lui and Baldwin, 2012) to remove parallel sentences where the identified language does not match the expected one. Data amounts after filtering are shown in the final column of Table 1. Similar to the amount of duplicates, AMI and jParaCrawl were filtered the most along with the WMT data set.

For pre-processing we used only Sentencepiece (Kudo and Richardson, 2018) to create a shared vocabulary with size depending on the total training data set size for the specific experiment. The vocabulary size was set to 3,000 tokens for experiments with only BSD 20 data, 8,000 for experiments with BSD 80 data, 16,000 when using BSD 80 / AMI / ON together and 32,000 tokens for experiments involving WMT, jParaCrawl or News data. We did not perform other tokenisation or truecasing for the training data. We used Mecab (Kudo, 2006) to tokenise the Japanese side of the evaluation data, which we used only for scoring. The English side remained as-is.

3 Model Configurations

We separate our submissions in 3 main categories by model configuration type - statistical MT (SMT) baseline models, NMT models and NMT models with context. The latter category also includes models with tags specifying the domain (or rather training corpus used).

3.1 SMT

We trained SMT baseline systems using using the Moses (Koehn et al., 2007) toolkit in the Tilde MT platform (Vasiļjevs et al., 2012). The SMT systems consist of: word alignment performed using fast-align (Dyer et al., 2013); 7-gram translation models and the *wbe-msd-bidirectional-fe-allff* reordering models; a language model trained with KenLM (Heafield, 2011); models tuned using the improved MERT (Bertoldi et al., 2009).

3.2 NMT

For the sentence-level NMT systems, we used Sockeye (Hieber et al., 2017) or Marian (Junczys-Dowmunt et al., 2018) to train transformer architecture (Vaswani et al., 2017) models with several different parameter configurations until convergence on development data (no improvement on validation perplexity for 10 checkpoints). Each model was trained on a single Nvidia TITAN V (12GB) GPU, and training time was about 2-3 days for models with only BSD/ON/AMI data and about 5-6 days when using WMT/News/jParaCrawl data.

The main reason for using two different toolkits is that Marian currently does not support source side input factors, which help when training models with context. However, Sockeye does not support using optimiser delay, which enables training with larger batch sizes and significantly improves the final outcome. Differences in the model and data configurations are as follows:

- Sockeye
  - Transformer base (T.bas) - 6 layers
  - Transformer small (T.sm) - 4 layers
  - One previous context sentence (Ctx)
  - Domain tags (Dom)
  - Average of 4 best models (Avg)
- Marian
  - Transformer base (T.bas) - 6 layers
  - Optimiser delay of 8
  - Domain adaptation (Tun)
  - Ensemble of 2 best models (Ens)

We experimented with two different approaches of domain adaptation. For models trained with Marian, the usual approach of resetting convergence parameters and swapping out the full training data set with a 1:1 mix of domain data (BSD corpus) and an equal-sized random subset of the remaining data worked fine. This, however, did not work as well for models trained with Sockeye when following the domain adaptation tutorial[2]. In this case we augmented the training data by adding a domain specifying tag (*<AMI>*, *<BSD>* or *<ON>*) (Tars and Fishel, 2018) at the beginning of each source sentence of training, development and evaluation data. The domain tag approach lead to a similar increase in BLEU score as the usual domain adaptation approach.

[2] https://awslabs.github.io/sockeye/tutorials/adapt.html

3.3 NMT with Context

To train our context-aware systems, we experimented with the approach of sentence concatenation (Tiedemann and Scherrer, 2017) with source side factors (Sennrich and Haddow, 2016). We use the Sockeye toolkit and similar parameters as in our sentence-level systems. For the concatenation context-aware MT, we experimented with two approaches: 1) prepending the previous sentence from the same document, followed by a beginning of sentence tag <*bos*>, to the source sentence; 2) in addition, providing source side factors to specify if a token represents context or the source sentence.

The source side factors that we used for training were either C or S, representing context and the actual source sentence respectively. Examples of source sentences with context and factors are shown in Table 2. The first sentence in the table has no previous context, as it is the first one in the respective document. The second sentence has the first one as context, followed by a beginning of sentence tag <*bos*>, and so on.

Source sentences
<bos> はい、 G 社 お客様 相 談 室 の ケ イ ト です。
はい、 G 社 お客様 相 談 室 の ケ イ ト です。<bos> ご 用 件 は？
Source side factors
C S S S S S S S S S S S S S S
C C C C C C C C C C C C C C C S S S S S

Table 2: Examples of training data source sentences and the respective source side factors for the concatenated context-aware experiments.

4 Results

We use the SacreBLEU[3] tool (Post, 2018) to evaluate automatic translations and calculate BLEU scores (Papineni et al., 2002) in Table 3, which contains results from the intermediate models that were not submitted to the shared task evaluation site. This table shows the incremental BLEU score improvements of switching between the *base* and *small* configurations of the transformer model, model averaging, enabling optimiser delay and domain adaptation. It also shows that BLEU scores go both up and down when adding context sentences to the source side. We did not compare how data filtering impacts the final result, but filtering was only performed in experiment settings which involved the jParaCrawl corpus, which was the largest overall and contained the majority of noisy data.

Data Amount

Both result tables show that adding training data improves BLEU scores. Ideally, we would have wanted the jParaCrawl and all WMT corpora to be document-aligned to be able to train the context-aware models using the complete data set.

We first experimented with incrementally adding all of the document-level data available to us - BSD 80, AMI, ON, News - and compared how using context impacts the final translation. Then, we switched to only sentence-level experiments and added jParaCrawl and the rest of WMT20 corpora to the mix, which finally lead to our highest-scoring models.

Model Configurations

For experiments using only the provided training data from the shared task it is clear that the transformer-base model was too big to efficiently utilise the little amount of data. It is interesting that for EN→JA the SMT model outperformed all NMT models.

Rows 9-12 of Table 3 show incremental improvements while using the same training data and seemingly the same transformer-base configuration. We first switched from Sockeye to Marian and saw immediate improvement of about 1 BLEU. Later we found out that this was due to some default parameters being different or not set in Sockeye and after aligning the parameters[4] we were able to train comparable models. However, Sockeye does not support the optimiser delay feature that can be used in Marian to increase the effective training batch size and simulate training on larger GPUs, which in turn leads to higher final BLEU scores. Domain adaptation / tuning is another feature / strategy that supposedly works in both toolkits, but seems to lead to grater gain in Marian. Rows 11 and 12 show the improvement from optimiser delay and domain adaptation, which are about 1.6 BLEU and about 1.7 BLEU on average respectively.

[3] Version string: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.2.21

[4] Initial learning rate - 0.003; transformer activation type - swish; optimizer params - beta1:0.9, beta2:0.98, epsilon:0.000000001; transformer dropout - 0.1;

Configuration	EN→JA	JA→EN	Toolkit	Filtered
T.bas. | BSD 20	5.06	4.18	Sockeye	No
T.bas. | BSD 20 | Ctx	4.96	3.14	Sockeye	No
T.sm. | BSD 20	5.01	4.96	Marian	No
T.sm. | BSD 20	6.37	7.16	Sockeye	No
T.sm. | BSD 20 | Avg	6.49	7.22	Sockeye	No
T.sm. | BSD 20 | Ctx | Avg	7.23	6.93	Sockeye	No
T.sm. | BSD 80 | Ctx	12.39	14.07	Sockeye	No
T.bas. | BSD 80 | Ctx	12.74	14.92	Sockeye	No
T.bas. | BSD 80 / jParaCrawl / AMI / ON	14.12	18.24	Sockeye	Yes
T.bas. | BSD 80 / jParaCrawl / AMI / ON	15.16	19.38	Marian	Yes
T.bas. | BSD 80 / jParaCrawl / AMI / ON | Delay	16.71	21.16	Marian	Yes
T.bas. | BSD 80 / jParaCrawl / AMI / ON | Delay | Tun	19.32	22.85	Marian	Yes
T.bas. | BSD 80 / News / WMT / AMI / ON | Delay | Tun	19.23	23.25	Marian	Yes
T.bas. | BSD 80 / WMT / AMI / ON | Delay | Tun	19.56	22.97	Marian	Yes

Table 3: Automatic evaluation results of models that were not submitted to the shared task evaluation site. All EN→JA scores are calculated on references and outputs tokenised with Mecab. Configuration details are split by vertical lines, where the first part specifies the model type (Transformer - small or base), next are the corpora used for training, following by additional data/model details (domain tags, context, optimiser delay, domain adaptation, model averaging).

Context

By prepending the previous sentence as context for each training, development, and test data content sentence we were expecting to see slight improvements in both translation directions. We did, however, find that this leads to a drop in scores for all of our JA→EN experiments (rows in Tables 3 and 4 where the difference between adjacent configurations is *Ctx*). Out of the 5 comparable EN→JA experiments adding context improved in 3 cases.

4.1 Automatic Evaluation

Automatic evaluation results from the submission website are shown in Table 4. The abbreviations used in the table are explained in Section 3.2. Several of our models ranked in the top-5 in each translation direction according to the automatic evaluation. By looking at the results, it is clear that having the larger BSD corpus gave us a big and perhaps unfair advantage. It is also evident both here and in the human evaluation results that just adding larger amounts of any parallel data leads to improvements in BLEU scores.

4.2 Human Evaluation

Results of the human evaluation (Nakazawa et al., 2020) are summarised in Table 5. For the human evaluation we chose to submit our highest-scoring context-aware systems along with their otherwise identical context-agnostic alternatives in order to better understand the benefits or drawbacks of adding previous context. We added all human evaluated results to the table and gathered configurations of other team models from descriptions on the evaluation site[5].

Unlike BLEU and RIBES scores, which were higher for the context-aware version in the EN→JA direction, it seems that the evaluators preferred the context-agnostic model output in both translation directions.

We were also fortunate enough to have our overall highest-scoring submissions evaluated by humans and confirm that they truly were in the top-2 for both translation directions.

5 Conclusion

The paper described the development process of the The University of Tokyo's MT systems that were submitted for the WAT 2020 Document-level Business Scene Dialogue Translation sub-task. Among other things, we experimented with adding previous context to training data, larger batches and domain specifying tags. While we did find some slight BLEU score improvements when training context-aware models, document-aligned data required to train them are still rare and rather small in size. More substantial improvements were gained by simply adding all available sentence-aligned

[5] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/index.html

Configuration	EN→JA		JA→EN	
	BLEU	RIBES	BLEU	RIBES
SMT | BSD 20	10.18	64.48	6.88	57.53
T.sm. | BSD 20	7.88	59.93	7.67	58.08
T.sm. | BSD 20 | Ctx | Avg	8.89	61.96	7.43	57.63
T.sm. | BSD 80	14.73	70.38	15.83	70.00
T.sm. | BSD 80 | Ctx	14.27	70.69	14.49	69.70
T.bas. | BSD 80 / WMT / AMI / ON	14.54	66.25	18.94	69.81
T.bas. | BSD 80 / AMI / ON	16.35	70.18	17.58	71.07
T.bas. | BSD 80 / AMI / ON | Dom | Avg	16.67	71.84	18.57	72.08
T.bas. | BSD 80 / AMI / ON | Dom / Ctx | Avg	17.24	73.22	18.05	72.32
T.bas. | WMT	17.18	71.00	16.99	67.71
T.bas. | BSD 80 / News / AMI / ON | Ctx	19.80	**74.63**	21.64	**74.60**
T.bas. | BSD 80 / jParaCrawl / AMI / ON | Tun	21.24	73.74		
T.bas. | BSD 80 / News / WMT / AMI / ON | Tun | Ens	**21.85**	**74.13**		
T.bas. | BSD 80 / WMT / AMI / ON | Tun | Ens			**23.80**	**74.69**

Table 4: Automatic evaluation results of the submitted systems in BLEU and RIBES. All EN→JA scores are an average of the 3 tokeniser versions (Juman, Kytea and Mecab). The first two groups of rows were trained with Sockeye and the last group was trained with Marian. Configuration details are split by vertical lines, where the first part specifies the model type (SMT or Transformer - small or base), next are the corpora used for training, following by additional data/model details (domain tags, context, domain adaptation), and finally if either model averaging (only for Sockeye) or ensembling (only for Marian) was used. Configurations marked in a bold font were submitted for human evaluation.

Configuration	EN-JA		JA-EN		Team
	BLEU	Human	BLEU	Human	
T.bas. | News / WMT / BSD 80k / AMI / ON | Tun | Ens	21.85	4.23			ut-mrt
mBART pre-training | JESC | Doc-lvl | Ens	22.07	4.20	23.15	4.19	goku20
T.bas. | WMT / BSD 80k / AMI / ON | Tun | Ens			23.80	4.12	ut-mrt
T.bas. | BSD 20k / JESC / KFTT / MTNT / + | BT | Tun | Ens	22.31	4.13	22.83	4.10	DEEPNLP
T.bas. | BSD 20k / JESC / OpenSubtitles | Tun			18.70	3.93	adapt-dcu
T.bas. | BSD 80k / AMI / ON | Dom | Avg	16.67	3.56	18.57	3.62	ut-mrt
mBART pre-training | Doc-lvl | single model	17.04	3.55	17.02	3.57	goku20
T.bas. | BSD 80k / AMI / ON | Dom / Ctx | Avg	17.24	3.52	18.05	3.55	ut-mrt
T.bas. | BSD 20k | Ens	11.29	2.60	10.91	2.40	DEEPNLP

Table 5: Human evaluation results ordered by the human adequacy score on a scale of 0.00 to 5.00 - the higher the better. All EN→JA BLEU scores are an average of the 3 tokeniser versions (Juman, Kytea and Mecab). In addition to the previously introduced abbreviations, BT stands for back-translation, Doc-lvl means document level, the + signifies other unmentioned corpora that were used, and the remaining abbreviations are corpora that the other shared task participants used.

corpora and training regular NMT models.

In contrast to our expectation that the context-aware models will be superior at least for the EN→JA translation direction, where we saw gains in BLEU scores, results from the human evaluation showed otherwise. We believe that a more sophisticated training method may be required to fully take advantage the document-aligned data.

We did not perform any back-translation of monolingual business dialogue or similar corpora, nor did we train *transformer-big* models or perform model distillation. All of these are other popular methods used in similar shared tasks known to improve the final results. Our intuition is that such moves would further improve the final outcome by several BLEU points, but due to time constraints we chose not to go forward with them. In total, 26 systems were submitted for the English↔Japanese language pair and four of them to the human evaluation.

Acknowledgements

This work was supported by "Research and Development of Deep Learning Technology for Advanced Multilingual Speech Translation", the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN.

References

Nicola Bertoldi, Barry Haddow, and Jean-Baptiste Fouet. 2009. Improved Minimum Error Rate Training in Moses. *The Prague Bulletin of Mathematical Linguistics*, 91(1):7—–16.

Nikolay Bogoychev, Kenneth Heafield, Alham Fikri Aji, and Marcin Junczys-Dowmunt. 2018. Accelerating asynchronous stochastic gradient descent for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2991–2996, Brussels, Belgium. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of NAACL-HLT 2013*. 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. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *ArXiv e-prints*.

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.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. jp*.

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.

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. Association for Computational Linguistics.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2019. Jparacrawl: A large scale web-based english-japanese parallel corpus. *arXiv preprint arXiv:1911.10668*.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2020. Overview of the 7th workshop on Asian translation. In *Proceedings of the 7th Workshop on Asian Translation*, Suzhou, China. Association for Computational Linguistics.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz. 2018. JESC: Japanese-English Subtitle Corpus. *Language Resources and Evaluation Conference (LREC)*.

Matīss Rikters. 2018. Impact of Corpora Quality on Neural Machine Translation. In *In Proceedings of the 8th Conference Human Language Technologies - The Baltic Perspective (Baltic HLT 2018)*, Tartu, Estonia.

Matīss Rikters, Ryokan Ri, Tong Li, and Toshiaki Nakazawa. 2019. Designing the business conversation corpus. In *Proceedings of the 6th Workshop on Asian Translation*, pages 54–61, Hong Kong, China. Association for Computational Linguistics.

Matīss Rikters, Ryokan Ri, Tong Li, and Toshiaki Nakazawa. 2020. Document-aligned japanese-english conversation parallel corpus. In *Proceedings of the Fifth Conference on Machine Translation: Volume 1, Research Papers*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. In *Proceedings of the Twenty-first Annual Conference of the European Association for Machine Translation (EAMT 2018)*.

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.

Andrejs Vasiļjevs, Raivis Skadiņš, and Jörg Tiedemann. 2012. LetsMT!: A Cloud-Based Platform for Do-It-Yourself Machine Translation. In *Proceedings of the ACL 2012 System Demonstrations*, July, pages 43–48, Jeju Island, Korea. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł 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.